МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АНАЛІЗУ І ПРИЙНЯТТЯ РІШЕНЬ ПРИ УПРАВЛІННІ ІТ-ПРОЕКТАМИ»

Здобувача вищої освіти другого (магістерського)
рівня,
ІІ курсу, групи КМІ-22-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології в економіці»

Пономаренко Андрій Валерійович

Науковий керівник Ясенова Ірина Сергіївна, к.т.н., доц.

Керівник освітньо-професійної програми Горбовий Артур Юліанович, д.т.н, професор

Допущено до захисту за рішенням кафедри
протокол від «28» грудня 2023 № 8

Завідувач кафедри Філоненко Михайло Миколайович, к.ф.-м.н., доцент

Ірпінь – 2024

Реєстрація №13 29.01.2024 _____
(дата) (підпис відповідальної особи)

Результати перевірки до захисту 29.01.2024 _____
(до захисту, на доопрацювання) (дата) (підпис)

Результат захисту 4 (добре) 12.02.2024 _____
(оцінка) (дата) (підпис)

Голова комісії: _____ В.М. Франчук
(підпис)

Члени комісії: _____ А.Ю. Горбовий
(підпис)

_____ О.В. Редич
(підпис)

_____ В.О. Ніжегородцев
(підпис)

БЛАНК ОЦІНЮВАННЯ ВИПУСКНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії,секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	25		12.02.2024 Франчук В.М. Горбовий А.Ю. Редич О.В. Ніжегородцев В.О. Суховецька Л.П.
2. Дотримання вимог щодо оформлення	від 0 до 10	7		
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	3		
4. Рецензент	від 0 до 5	5		
РАЗОМ БАЛІВ	від 0 до 50	40		

Оцінювання кваліфікаційної роботи науковим керівником - 40 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалоюЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
80	B	4 (добре)	12.02.2024	Франчук В.М. Горбовий А.Ю. Редич О.В. Ніжегородцев В.О. Суховецька Л.П.

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	3 (задовільно)
50-59	E	
35-49	FX	2 (незадовільно)

Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: «Інформаційні управляючі системи і технології в економіці»

Затверджую
Завідувач кафедри _____ Філоненко М.М.
«04» 04 2023

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Пономаренку Андрію Валерійовичу

1. Тема кваліфікаційної роботи
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АНАЛІЗУ І ПРИЙНЯТТЯ РІШЕНЬ ПРИ УПРАВЛІННІ ІТ-ПРОЕКТАМИ
Затверджена наказом ректора від «04» квітня 2023 № 362
2. Строк здачі закінченої роботи «20» грудня 2023
3. Мета роботи полягає у розробці та аналізі програмного забезпечення для системи аналізу і прийняття рішень при управлінні ІТ-проектами. Основними завданнями є створення ефективного інструментарію для моніторингу та аналізу процесів в ІТ-проектах, вдосконалення механізмів прийняття управлінських рішень на основі отриманих даних, а також підвищення загальної ефективності управління проектами за допомогою автоматизованих засобів. Результати роботи покликані сприяти підвищенню якості управління ІТ-проектами через вдосконалення програмного забезпечення для аналізу та прийняття рішень.

Об'єкт дослідження: Управління ІТ-проектами

Предмет дослідження: Програмне забезпечення для системи аналізу і прийняття рішень при управлінні ІТ-проектами.

Вихідні дані до роботи: статистика успішності та невдач ІТ-проектів, інформація про сучасні методи управління та їхню ефективність, дані щодо витрат, тривалості та інших параметрів ІТ-проектів, специфікації щодо функціональності системи аналізу та прийняття рішень, вимоги щодо безпеки та конфіденційності даних.

4. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)

Розділ 1. Аналіз процесу планування проектів розробки програмного забезпечення (аналіз основних методик ведення проектів, аналіз основних методів прогнозування термінів проекту, огляд і аналіз систем ведення проекту). Розділ 2. Аналіз інформаційної структури процесу розробки програмного забезпечення, функціонально-інформаційна модель процесу виконання ІТ-проекту, вимоги до системи підтримки прийняття рішень при плануванні і виконанні проекту. Розділ 3. Розробка програмного забезпечення.

5. Календарний план виконання роботи

№ п/п	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 27.12.2022	До 27.12.2022
2	**І розділ** (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	26.06.2023	26.06.2023
3	**ІІ розділ** (аналіз експериментальної частини про методики досліджень)	23.10.2023	23.10.2023
4	**ІІІ розділ** (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	30.11.2023	30.11.2023
5	**Висновки**	10.12.2023	10.12.2023
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	20.12.2023	20.12.2023
7	Отримання відгуку керівника	29.01.2024	29.01.2024
8	Отримання зовнішньої рецензії	01.02.2024	01.02.2024

6. Дата видачі завдання «04» квітня 2023

7. Керівник випускної кваліфікаційної роботи _____ (підпис) Ясенова І.
(прізвище, ініціа

8. Керівник освітньо-професійної програми _____ (підпис) Горбовий
(прізвище, ініціа

9. Завдання до виконання отримав здобувач вищої освіти _____ (підпис) Пономарен
(прізвище, ініціа

АНОТАЦІЯ

Дипломна робота присвячена розробці та впровадженню програмного забезпечення для системи аналізу та прийняття рішень при управлінні IT-проектами. У роботі детально розглядаються основні аспекти управління проектами в області інформаційних технологій та визначаються ключові етапи цього процесу. Зосереджуючись на важливості ефективного прийняття рішень, автор вивчає існуючі методології та інструменти для аналізу та оптимізації проектів.

Мета роботи полягає у розробці та аналізі програмного забезпечення для системи аналізу і прийняття рішень при управлінні IT-проектами. Основними завданнями є створення ефективного інструментарію для моніторингу та аналізу процесів в IT-проектах, вдосконалення механізмів прийняття управлінських рішень на основі отриманих даних, а також підвищення загальної ефективності управління проектами за допомогою автоматизованих засобів.

Об'єкт дослідження: Управління IT-проектами

Предмет дослідження: Програмне забезпечення для системи аналізу і прийняття рішень при управлінні IT-проектами.

Ключовими аспектами роботи буде розробка імовірностних моделей для прогнозування ризиків та визначення оптимальних стратегій управління, інтеграція інструментів збору та аналізу даних для підвищення точності прийняття рішень, а також створення зручного і інтуїтивно зрозумілого інтерфейсу для користувачів системи.

Крім того, дипломна робота має на меті дослідження сучасних тенденцій у сфері управління IT-проектами та визначення важливих факторів, що впливають на успішність проектів. Результати роботи покликані сприяти підвищенню якості управління IT-проектами через вдосконалення програмного забезпечення для аналізу та прийняття рішень.

За результатами роботи було досліджено ефективність розробленої системи на практиці, використовуючи реальні IT-проекти як об'єкт дослідження. Дипломна робота розкриває необхідність та переваги використання програмного забезпечення для управління IT-проектами та детально розробляє та впроваджує систему, спрямовану на покращення процесів прийняття рішень в цій області.

Кваліфікаційна робота містить 81 сторінку, 22 рисунки, список використаних джерел з 50 найменувань.

ABSTRACT

The diploma thesis is dedicated to the development and implementation of software for the analysis and decision-making system in IT project management. The paper extensively examines the fundamental aspects of project management in the field of information technology and defines the key stages of this process. Focusing on the importance of effective decision-making, the author studies existing methodologies and tools for project analysis and optimization.

The objective of the thesis is the development and analysis of software for the analysis and decision-making system in IT project management. The main tasks include creating an efficient toolkit for monitoring and analyzing processes in IT projects, improving decision-making mechanisms based on obtained data, and enhancing overall project management efficiency through automated tools.

Research Object: IT Project Management

Research Subject: Software for the analysis and decision-making system in IT project management.

Key aspects of the work involve developing probabilistic models for risk prediction and determining optimal management strategies, integrating data collection and analysis tools to improve decision accuracy, and creating a user-friendly and intuitive interface for system users.

Additionally, the thesis aims to explore current trends in IT project management and identify significant factors influencing project success. The results of the work are intended to contribute to the improvement of IT project management quality through the enhancement of software for analysis and decision-making.

The effectiveness of the developed system was investigated in practice using real IT projects as the research object. The thesis highlights the necessity and advantages of using software for IT project management and comprehensively develops and implements a system aimed at improving decision-making processes in this field.

The qualification work comprises 81 pages, 22 figures, and a bibliography with 50 references.

ЗМІСТ

ВСТУП 8

РОЗДІЛ 1. АНАЛІЗ ПРОЦЕСУ ПЛАНУВАННЯ ПРОЕКТІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 10

1.1 Визначення поняття проекту 10

1.2 Аналіз процесу планування проектів 11

1.3 Аналіз основних методик ведення проектів 13

1.4 Аналіз основних методів прогнозування термінів проекту 19

1.5 Огляд і аналіз систем ведення проекту 25

1.6 Висновки 30

РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОЇ СТРУКТУРИ ПРОЦЕСУ ВИКОНАННЯ ПРОЕКТУ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 32

2.1 Загальна схема і етапи виконання проекту 32

2.2 Функціонально-інформаційна модель процесу виконання ІТ-проекту 35

2.3 Перетворення даних в процесі виконання ІТ-проекту 41

2.4 Автоматизація процесу розробки ПЗ 45

2.5 Вимоги до системи підтримки прийняття рішень при плануванні і виконанні проекту 49

2.6 Висновки 52

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 53

3.1 Архітектура системи підтримки прийняття рішень при плануванні проекту 53

3.2 Інтерфейс системи підтримки прийняття рішень 62

3.3 Прогнозування процесу виконання завдань в проекті розробки системи автоматизованого аналізу текстів 65

3.4 Подальше використання розробленої системи підтримки прийняття рішень 74

3.5 Висновки 75

ВИСНОВОК 76

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 78

ВСТУП

Актуальність теми

При розробці програмного забезпечення (ПЗ) велике значення має планування проекту. Часто саме через помилки в плануванні програмний продукт не випускається в термін або НЕ укладається в виділений бюджет. З одного боку, це пов'язано зі складністю завдання планування і врахування особливостей продукту, замовника, команди розробки, фінансових коштів і т.д., з іншого - з великою роллю людського чинника в питаннях планування.

Інтенсивний розвиток інформаційних і комунікаційних технологій призвів до формування нових вимог до планування та управління проектами в різних областях. Одним з обов'язкових умов ефективної реалізації проектів стає застосування сучасних засобів та інструментів управління проектами, заснованих на використанні нових інформаційних технологій. Розвиток спеціального ПЗ для планування і управління проектами обумовлено, в першу чергу, необхідністю максимальної інтеграції найбільш ефективних методів, засобів і інструментів теорії управління проектами для оперативного доступу до аналітичними даними за станом проекту.

Існуючі методики планування проекту пропонують різні підходи до побудови процесу розробки ПЗ («каскадний», ітераційний, гнучкий), розроблені програмні засоби для ведення проектів в рамках тієї або іншої методики (MS Project, Primavera Project Planner, Spider Project і т . д). Їх головним недоліком є те, що помилки в плануванні можуть бути враховані вже тільки після завершення проекту. Оскільки процес розробки дуже динамічний, вимоги і набір завдань можуть часто змінюватися, помилка в плануванні може привести до значної затримки часу закінчення проекту і / або істотної перевитрати ресурсів. Існуючі методи попередньої оцінки складеного плану проекту (PERT , COCOMO) вимагають завдання для ще невідомого проекту великої кількості характеристик і найчастіше дають недостатньо точні результати.

З огляду на це актуальним є завдання аналізу інформаційної структури і створення спеціального алгоритмічного забезпечення системи аналізу та прийняття рішень при плануванні процесу розробки, яка дозволить до початку реалізації проекту і в процесі його виконання отримувати оцінку ймовірності успішного виконання проекту за певний термін.

Метою роботи є підвищення точності оцінки часу необхідного для виконання IT-проектів шляхом аналізу інформаційної структури і розробки алгоритмічного забезпечення системи аналізу та підтримки прийняття рішень при управлінні проектами розробки ПЗ.

Для досягнення поставленої мети в роботі вирішені наступні завдання:

• аналіз відомих моделей процесу розробки ПЗ , підходів до планування та існуючих програмних продуктів для ІТ-проектів ;

• дослідження інформаційних процесів при виконанні ІТ-проекту з метою проведення моделювання та їх автоматизації;

• розробка функціонально-інформаційної моделі процесу виконання ІТ-проекту з використанням ймовірнісної моделі планування;

• перевірка створених моделей і системи підтримки прийняття рішення при плануванні ІТ-проекту .

Об'єкт дослідження

Об'єктом дослідження є процес розробки ПЗ .

Предмет дослідження

Предметом дослідження є методи планування ІТ-проектів і методики прийняття рішень при управлінні проектами розробки ПЗ.

РОЗДІЛ 1. АНАЛІЗ ПРОЦЕСУ ПЛАНУВАННЯ ПРОЕКТІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Визначення поняття проекту

Проект зазвичай визначається як унікальний набір скоординованих дій із заданим початком і завершенням, здійснюваних індивідуумом або організацією для вирішення специфічних завдань відповідно до наявного розкладу, витратами і параметрами виконання [85, 62].

Головна властивість проекту в тому, що він повинен мати обмеження в часі. Унікальність кожного проекту породжує складності при його плануванні, оскільки часто важко достовірно припустити, як будуть досягатися результати і наскільки вдасться вкластися в заздалегідь визначені терміни. Результати проектної діяльності є виконані завдання і набутий досвід, який враховується при плануванні наступних проектів.

Управління проектами складається з двох основних частин: складання плану і контроль ходу робіт по ньому. Якісний план проекту полегшує в подальшому виконання проектних завдань і підвищує ймовірність вдалого завершення проекту. Є безліч чинників, які позитивно і негативно впливають на хід проекту. Наприклад, декомпозиція цілей на більш дрібні і більш зрозумілі зазвичай позитивно впливає на хід проекту [1, 21, 24- 25, 32-35, 46-47, 71-72, 83-86, 91].

Управління проектами - це сукупність процесів (методик, моделей, програмних і технічних засобів, методологій), які здійснюються при розробці та реалізації проектів, процеси ці мають обмеження в часі, вимагають витрат ресурсів.

Під ресурсами проекту в літературі зазвичай розуміють бюджет, час, працю, матеріали, енергію, і т.д. У області розробки програмного забезпечення являють ся істотними:

- тимчасові ресурси;
- людські ресурси;
- фінансові ресурси (бюджет).

При плануванні проводиться загальний аналіз робіт і ресурсів, також необхідно враховувати їх обмеженість і передбачуване ра розподілом, на основі потреб проектів. Ресурсне планування включає в себе аналіз ресурсів і робіт, розробку системи розподілу ресурсів, контроль над ходом робіт, вибір коригувальних заходів [47].

У кожного проекту чітко визначено початок і критерії закінчення. Проект закінчується разом з досягненням всіх його цілей або коли ці цілі не

можуть бути досягнуті. Проекти зазвичай мають чітко окреслені часові рамки для створення продукту, часто встановлюються ринком [7, 87].

1.2 Аналіз процесу планування проектів

1 .2.1 Визначення мети проекту

Реалізація проекту розробки програмного забезпечення відповідно до пред'явлених вимог - вкрай актуальне завдання. Проект починається з визначення мети, яку, найчастіше неможливо сформулювати на початку проекту. Це призводить до того, що учасники проекту розуміють її по-різному. В результаті мета проекту уточнюється в міру виконання проекту, що в підсумку призводить до змін у вимогах по ходу проек та [23].

Потім зазвичай складається план в традиційному розумінні, де планування полягає в призначенні робіт, визначення їх тривалості та бюджету. Подальший контроль виконання полягає в порівнянні фактичної тривалості робіт і витрат з пл Анів значеннями. При цьому цілі проекту, заради досягнення яких проект власне і задумувався, відходять на другий план, і про них згадують тоді, коли проект майже завершений [1, 19, 65].

1 .2.2 Аналіз проблеми планування ресурсів проєкта

На початку кожного проекту стоїть завдання визначення його ресурсоємності, часу реалізації, бюджету і відповідно здійсненності при обраному поєднанні цих параметрів. Це завдання вирішується наближено, оскільки існуючі методики пропонують тільки приблизні оцінки, практично не враховують специфіку проекту, а необхідні ресурси визначаються виключно на основі досвіду і суб'єктивної думки людей, які виступають в ролі експертів [1], не завжди є фахівцями в даній області [46].

У будь-якому проекті доводиться балансувати між вартістю, часом, якістю та обсягом реалізованої функціональності. Точний розрахунок ресурсів, необхідних для реалізації даного продукту за певних вимог до якості, явля ється однією з основних проблем в галузі управління проектами. Особливо якщо мова йде про проекти розробки програмного забезпечення. При цьому багатьма авторами виділяється, в першу чергу, велика творча складова процесу розробки програмного забезпечення [20, 64-66]. Тому в розрахунках характеристик проектів виникають складності обліку величезної кількості чинників, в тому числі трудноформалізуемих, що впливають на життєвий шлях проекту. Як наслідок, сьогодні багато компаній стикаються з серйозним і проблемами в разі неправильних розрахунків необхідних термінів [77]:

• при недооцінці - непередбачена витрата додаткових коштів, невдоволення замовника невиконанням зобов'язань в обумовлений час,

переробки співробітників, складність управління некон троліруемим проектом, низька якість кінцевого продукту, недостатні функції системи і т. Д .;

• при переоцінці - перевитрата витрата ресурсів, залучених до проекту, відмова замовника від контракту з даними умовами і т.д.

Точні оцінки витрат вироб ництва програмного забезпечення важливі як розробникам, так і замовнику. Виникає потреба в розробці методів і засобів, що дозволяють менеджеру проекту оцінити необхідні тимчасові і людські ресурси на основі всіх наявних даних: історії попе дущіх подібних проектів, досвіду та продуктивності праці співробітників, специфіки компанії і т.п. Крім того, потрібно також мати можливість перерахунку і уточнення термінів і ресурсів вже в ході розробки програмних систем з урахуванням розбіжності поточного стану проекту і його плану. Це мо жет допомогти керівнику своєчасно виявити відхилення від встановленого графіка і вжити відповідних заходів в управлінні проектом [28].

1.3 Аналіз основних методик ведення проектів

1.3.1 Трикутник взаємозалежностей частин проекту

Більшість проектів мають певні дату закінчення, бюджет і обсяг робіт. Час, ресурси і обсяг робіт в літературі [8, 41-42, 56-57] зазвичай називають *проектним трикутником* (рисунок 1.1), тому що зміни в одному з цих елементів викликають зміни в інших. У загальному випадку, для проекту важливі всі три складові частини, але, як правило, тільки один з них в залежності від пріоритетів має найбільший вплив на інші. У цьому випадку говорять про проекти з фіксованим терміном виконання (time - driven , сам им істотним параметром є час), з фіксованим обсягом робіт (feature driven , для реалізації основних функцій системи можуть бути збільшені ресурси або терміни) і з фіксованою вартістю (resource driven , пріоритетним є обмеження одне одного з ресурсів) [60, 64].

Наприклад, якщо за рахунок зменшення термінів окремих робіт змінюється план проекту, то зростає вартість проекту (якщо, наприклад, вирішено залучити додаткових працівників) або повинен зменшитися обсяг робіт. Якщо ж змінити план проекту з метою зменшення його бюджету, то може зрости тривалість проекту і зменшитися обсяг робіт. Нарешті, якщо збільшити обсяг робіт, то проект буде тривати довше і коштувати дорожче. Те, як зміни в плані впливають на інші сторони трикутника, залежить від обставин та специфіки проекту [75].

Якість, четвертий елемент проектного трикутника, знаходиться в його центрі, і зміни, які вносяться в будь-яку зі сторін трикутника, практично завжди впливають на якість (рисунок 1.2). Якість не є стороною трикутника - це

результат дій з часом, вартістю та обсягом робіт [43]. У роботах деяких авторів трикутник за рахунок д Додавання якості перетворюється в піраміду [68, 76].



Рис. 1.1 - Проектний трикутник

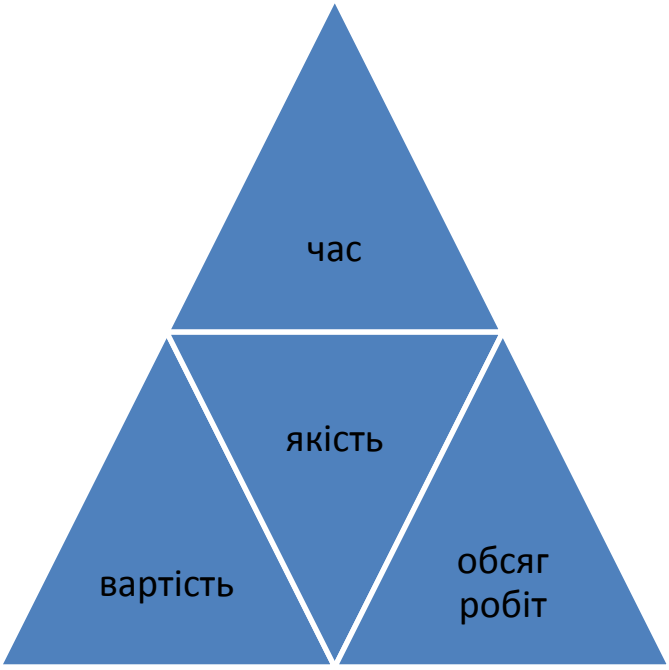


Рис. 1.2 - Якість в проектному трикутнику

1.3.2 Огляд методик ведення проектів на основі обмежень

У літературі наводиться безліч способів класифікування підходів до розробки програмного забезпечення. Одним з перших найбільш повний описів може вважатися [11]. У стандарті ДСТУ ISO / IEC 12207-1999 [17] згадується каскадна, спіральна і еволюційна моделі. У новій версії стандарту кількість моделей зростає до чотирьох (додається итерационная модель). У наступних же

версіях декларується безліч варіантів організації процесу розробки ПЗ [18], але основними є: *каскадний, інкрементна і еволюції іонна*.

У роботах [3, 5, 9-10, 54, 58-59, 89] розглядається безліч сучасних варіантів організації розробки програмного забезпечення, виробляються критерії оцінки та вибору альтернатив, а також і їх різні класифікації. Однак для більшінс тва сучасних варіантів побудови процесу розробки програмного забезпечення точна класифікація ускладнена. Практично кожен проект, часто намагаючись слідувати тим чи іншим принципам, адаптує їх до своїх реалій, тим самим створюючи новий підхід. При цьому практично завжди команда розробки в чому-небудь обмежена. Тому в роботі ці підходи класифікуються за способом побудови процесу розробки програмного забезпечення на основі пріоритетного типу обмежень.

У ибор серед варіантів побудови процесу розробки ПЗ являє собою досить складну задачу. Найчастіше такий вибір відбувається без урахування всіх можливих альтернатив, розглядаючи тільки обмежений набір методик, при цьому не враховується специфіка про єкта.

Підходи до розробки ПЗ, засновані на обмеження розмірів проекту, історично виникли першими. Це пояснюється в першу чергу тим, що обмеження розмірів проекту (як, втім, і ресурсів) логічно випливають з самої суті проекту. Як тільки вдається визначити і декомпозировать мета проекту відразу виникають рамки проекту (*scope)*, в яких визначено проект і обмеження. Проект, який не має меж, заздалегідь неуспешен. Всі підходи даної групи засновані на чіткому розподілі проекту на фази, відмінність аясь тільки тривалістю і ступенем опрацювання фаз.

Існує набір стандартів, що визначають різні елементи в структурі життєвих циклів програмного забезпечення та програмно-апаратних систем. В якості основних таких елементів виділяють технологич еские процеси - структуровані набори діяльностей, які вирішують деяку загальну задачу або пов'язану сукупність завдань, такі як процес супроводу ПЗ , процес забезпечення якості, процес розробки документації та ін. Процеси можуть визначати раз ні етапи життєвого циклу проекту і пов'язувати їх з різними видами діяльностей, артефактами і ролями зацікавлених осіб [14]. Найпоширенішими варіантами побудови процесу розробки ПЗ є: каскадний (водоспадний) підхід, V-обра зний, спіральний і ітераційний підходи.

Детальний розгляд цих підходів виходить за рамки даної роботи. Варто відзначити, що, не дивлячись на численну критику і великий вік каскадного підходу, часто, на пр актік, навіть якщо заявлений інший підхід, застосовується саме каскадний, по всій видимості, зважаючи на простоту його застосування.

Знаходження вірного балансу між ресурсами, часом розробки і можливостями - найважливіший момент в побудові рішення, що відповідає потребам замовника [8].

Наприклад, порівняння важливості обмежень кожного типу для підходів до розробки, дає діаграму, представлену на малюнку 1.3.

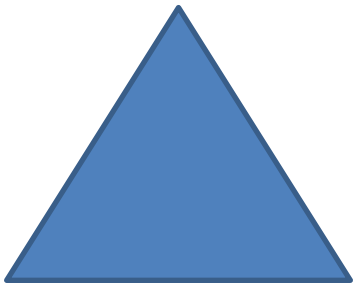
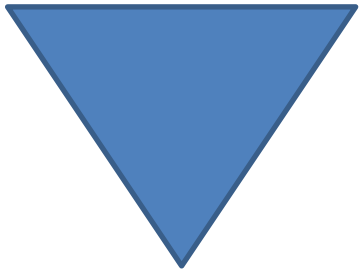
<table>
<tr><td></td><td colspan="2">Водоспадна, ітераційні моделі (роботи визначають вартість і терміни)</td><td colspan="2">Гнучкі методи (обсяг робіт визначається з обмежень по часу і вартості)</td></tr>
<tr><td>Обмеження</td><td colspan="2">Обсяг робіт</td><td>вартість</td><td>терміни</td></tr>
<tr><td></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>Оцінки</td><td>вартість</td><td>терміни</td><td colspan="2">Обсяг робіт</td></tr>
</table>

Рис. 1.3 -відмінності проектних трикутників для різних підходів

Методи побудови процесу розробки ПЗ , засновані на обмеженні ресурсів

У чистому вигляді даних методик розробки в ході дослідження виявлено не було. Однак обмеження різного виду ресурсів, безумовно, присутні в кожному проекті. Зазвичай вони слідують з обмежень решти типу, коли на стадії планування з'ясовується, яка зі складових буде грати найбільшими перепонами значення для проекту в цілому. Зазвичай, обмеження ресурсів доповнюють обмеження рамок проекту.

Методи побудови процесу розробки ПЗ , засновані на обмеженні часу.

Досвід останніх років показав, що для систем, вимоги до яких змінюються досить часто, необхідно максимально зменшити тривалість одного етапу жи зненного циклу. У зв'язку з цим в даний час стали дуже популярними

короткі життєві цикли розробки, які були об'єднані загальною назвою «гнучкі» методи (*agile* , від англ. «Гнучкий», «швидкий») [26-27, 61].

Найістотнішим обмеженням при під обном варіанті побудови процесу розробки ПЗ є час. Зазвичай на будь-якої короткий проміжок часу (1-3 тижні) фіксується набір функціоналу майбутньої системи, що підлягає реалізації. При цьому проектування, програмування, тестування в т ечень цього міні-циклу йдуть майже паралельно. Природно, не можна реалізовувати, а вже тим більше перевіряти модуль, який ще не спроектований.

Основна ідея життєвого циклу - максимальне укорочення тривалості кожного етапу життєвого циклу і тісно е взаємодія з замовником. По суті, на кожному етапі відбувається реалізація і тестування однієї функції системи, після завершення якого система відразу передається замовнику на перевірку або експлуатацію.

Головною проблемою є здійснення понятног про і документованого взаємодії, наприклад, інтерфейсами між модулями, що реалізують яку-небудь функцію. Якщо у всіх попередніх типах життєвого циклу інтерфейси досить чітко визначаються на самому початку розробки, оскільки заздалегідь відо стни всі модулі, то при «гнучкому» підході інтерфейси проектуються одночасно з розробляються модулями, що зазвичай робить негативний вплив на систему в цілому в довгостроковій перспективі.

Ще до появи прототипу прийнято приступати до створення тест овой середовища, максимально наближеної до реальної середовищі виконання програми, і коли з'являються проекти інтерфейсів - вже можна почати розробку прототипів функціональних тестів. Всі ці процеси, властиві всім моделям життєвого циклу, йдуть майже одночасно, що вимагає великої організованості і високого професіоналізму всієї команди розробки. Відсутність чіткої документованості процесів, незамінність кожного в команді, а також часто нестача сі стемного погляду на проект - найсерйозніші проблеми даного підходу. Але, не дивлячись на недоліки, Agile - підхід є найпрогресивнішим і найперспективнішим серед усіх існуючих, тому що він дозволяє подолати головний недолік всіх остальн их - статичність вимог і проекту в цілому. Програмне забезпечення повинно бути гнучким, змінюватися в залежності від зовнішніх умов і часто ніде не заявленим вимогам користувача.

Змішані підходи представляють найбільший інтерес для і сследованія і застосування різних методик розрахунку характеристик проектів, оскільки на практиці з огляду на безліч причин саме вони знаходять застосування. Головною причиною можна виділити унікальність кожного проекту і неможливість підібрати все маю щиеся умови під будь-якої один шаблон, тому

зазвичай говорять про пріоритетну використовуваної методикою і її адаптації для потреб конкретного проекту.

1.4 Аналіз основних методів прогнозування термінів проекту

План проекту, розроблений і затверджений до поч ала виконання робіт, є модель спрямованих на досягнення поставлених цілей взаємопов'язаних виробничих процесів. Як і будь-яка модель, календарний план повинен одночасно відповідати суперечливим умов простоти і досить й повноти. Для розробки календарних планів використовуються методи мережевого моделювання, які дозволяють пов'язати виконання різних робіт і процесів у часі, отримавши в результаті загальну тривалість всього проекту [2, 36]. В основу сучасного програмного забезпечення управління проектами (*Microsoft Office Project , Primavera , Open Plan , Spider Project)* також належить мережеве моделювання.

1.4.1 Планування мереж

П ри мережевому плануванні загальна пов лжітельность проекту залежить від взаємозалежності робіт, від продолжительностей робіт і тимчасових обмежень, встановлених на терміни робіт. Зазвичай при розрахунках розглядаються детерміновані мережеві моделі, взаємна послідовність і тривалості і яких задані однозначно; виділяється критичний шлях (роботи від точки старту до закінчення проекту, які більш не можуть бути розділені і виконуються максимальний час), на основі оцінок якого оцінюється проект цілком. Збільшення фактичної про тривалості робіт, що належать критичного шляху, по будь-якими причинами, призводить до відповідного збільшення загальної тривалості проекту, в той час як некритичні роботи мають деякі резервами часу [7, 22, 73-74]. Однак при уве личен фактичних термінів робіт некритичного шляху, варто враховувати можливість зміни критичного шляху і, як наслідок, поява необхідності повного повторного планування проекту. Одним з методів зниження ризику зсуву термінів закінчення робіт п ри детерминированном плануванні є виділення так званого страхового резерву часу, тобто плановий термін приймається на кілька відсотків (зазвичай 10-30%) менше договірного, а період часу, використовується для завершення невиконаних в строк робіт.

Внаслідок динамічного характеру і величезної кількості чинників, що впливають на успіх здійснення робіт, тільки в рідкісних випадках фактична тривалість і вартість робіт може збігтися з аналогічними параметрами, вказаний ими в календарному плані. Невизначеність, що має місце при плануванні, призводить до значних розбіжностей між планом і фактичним виконанням робіт і необхідності постійних перерахунків календарного плану. Також

непоодинокі випадки, при яких зрад яется запланована послідовність виконання робіт [81, 88].

Припустимо, що тривалість деякої роботи складно заздалегідь задати однозначно. Це може бути обумовлено багатьма причинами: низькою кваліфікацією виконавців, новизною робіт, наявністю недокументований частин або помилок в сторонньому ПЗ , недоліками в документації і т.д. У разі технічно складних проектів, пов'язаних з дослідженням нових технологій, однозначне визначення продолжительностей робіт, а також їх взаємозв'язків стає практично неможливим. Все сказане стосується не тільки до планування виконання робіт, але і до планування інших стадій IT-проекту : розробки документації, підготовці користувачів, налаштування обладнання і т.д. В даних випадках необхідно використовувати імовірнісні мережеві методи, застосування яких призводить, з одного боку, до ускладнення розрахунків, проте отримані результати є більш адекватними реальним умовам [16].

В даний час відомо безліч методів імовірнісного мережевого планування, найбільш поширеними з яких є: метод оцінки і аналізу програм *(Program Evaluation and Review Technique, PERT); м*етод статистичних випробувань або метод Монте-Карло; метод г рафіческой оцінки та аналізу програм *(Graphic Evaluation and Review Technique , GERT).*

1.4.2 Метод PERT

Метод оцінки та аналізу програм PERT був розроблений в 60-х роках минулого століття військовим відомством США, він заснований на логічній системі процесу, поставши ленній у вигляді спрямованого графа.

Особливість методу PERT полягає в можливості обліку імовірнісного характеру продолжительностей всіх або деяких робіт при розрахунку параметрів часу на мережевий моделі [31]. Він дозволяє визначати ймовірності закінчено ания проекту в задані періоди часу і до заданих термінах. Замість однієї детермінованою величини тривалості для робіт проекту задаються (як правило, експертним шляхом) три оцінки тривалості:

• **оптимістична** - робота не може бути виконана швидше, ніж за $t_{іопт}$;

• **песимістична** - робота не може бути виконана повільніше, ніж за $t_{іпесс)}$

• **найбільш ймовірна** $t_{і вір}$.

Потім імовірнісна мережева модель зводиться до детермінованої шляхом заміни трьох про ценок продолжительностей кожної з робіт однією величиною, званої очікуваною тривалістю t_{ie} , яка є середньозваженим трьох експертних оцінок тривалості даної роботи.

Для характеристики ступеня невизначеності оцінок розраховується середньоквадратичне відхилення і дисперсія оптимістичною і песимістичною

оцінки від середньозваженого значення. Враховуються тільки відхилення і дисперсії робіт , що належать критичного шляху.

Терміни виконання робіт і тривалість всього проекту можна розраховувати на мережевий моделі вже за допомогою детермінованих методів. Ймовірності завершення проекту в строки, що визначаються оптимістичною чи песимістичною величинами, прагнуть до нуля. Якщо підставити в детерміновану модель очікувані тривалості кожної роботи, то можна отримати загальну тривалість проекту як терміну, ймовірність закінчення до якого дорівнює 50%, тобто так звану очікувану пов лжітельность проекту [6, 44].

Метод PERT дозволяє отримати нормальний розподіл ймовірності тривалості, середнє значення якого відповідає очікуваної тривалості проекту. Для визначення ймовірності реалізації проекту за час, відмінне від очікуваного, розглядається величина середнє відхилення кривої нормального розподілу, яка відображає ступінь невизначеності оцінки тривалості всього проекту.

До переваг методу можна віднести його широку сферу застосування, фактично, він не прив'язаний до якої-небудь певної області і може бути застосований повсюдно. Також відносна простота і наочність розрахунків дозволяє його використовувати в даний час для пер первинних, грубих оцінок. Серед недоліків варто згадати раніше зазначену залежність від коректного визначення мережевого шляху, низька якість імовірнісних оцінок, слабкі можливості по оперативному реагуванню на зміни. Незважаючи на свої недоліки, метод PERT на поточний момент продовжує використовуватися при плануванні проектів.

1 .4.3 Метод Монте-Карло

Метод статистичних випробувань (метод Монте-Карло) полягає в розгляді мережі в якості ймовірнісної моделі, на якій оцінки продолжительностей окремих робіт можуть приймати будь-які значення, що лежать в крайніх (мінімум і максимум) зазначених експертами межах, і навіть виходити за ці межі з урахуванням розподілу ймовірностей. Сутність методу статистичних випробувань сос тоит в отриманні дуже великої кількості (близько десятків тисяч) окремих реалізацій даної мережевої моделі, що відрізняються один від одного тим, що тривалості робіт першій-ліпшій нагоді моделі випадково вибираються за законами, що характеризує ра розподілом, кожної з окремих оцінок продолжительностей. При цьому, хоча і з малою ймовірністю, будь-яка оцінка може навіть виходити за межі допуску, якщо тільки вся мережева модель «програна» досить велика кількість разів, так що поява значитель них відхилень окремих продолжительностей стало статистично можливим [80].

Для розрахунку мережевий моделі методом статистичних випробувань потрібні такі вихідні дані:

- перелік (номенклатура) робіт проекту;
- последоват ельность виконання робіт, яка характеризується встановленими зв'язками між роботами і задана на графі;
- функції щільності розподілу ймовірності тривалості кожної роботи.

Основними результатами застосування методу є:

• функція щільності розподілу ймовірності загальної тривалості проекту;

• функції щільності розподілу ймовірності ранніх і пізніх термінів початку і закінчення окремих робіт.

При кожному «випробуванні» мережевий моделі знаходиться критичний шлях і обчислюються всі необхідні параметри плану робіт. Загальна тривалість проекту, а також терміни виконання окремих робіт, визначені в результаті застосування методу, представляються у вигляді функцій розподілу Веро тності продолжительностей. На підставі даних функцій можна отримати дані, аналогічні результатам методу PERT . Метод статистичних випробувань є більш точним методом аналізу ймовірнісної мережевий моделі, зрозуміло, за умови коректності исхо дних даних. Серед недоліків методу можна виділити складну реалізацію, що вимагає перебудови для кожного проекту, а також складність ручного визначення статистичних функцій для кожної конкретної роботи [81]. *1.4.4 Модель планування проектів сосом / CO COMO II* Модель C OCOMO (COnstructive COst MOdel - модель витрат розробки) - це алгоритмічна модель оцінки інтегральної вартості, розробки ПЗ , розроблена Баррі Бьомом [69-70]. Модель використовує просту формулу регресії з параметрами, визначені ми з даних, зібраних по ряду проектів. Модель складається з ієрархії трьох послідовно деталізіруемая і уточнюються форм. Перший рівень - базовий, призначений для швидких ранніх оцінок вартості розробки ПЗ, має низьку точністю, внаслідок недо торих факторів, які неможливо врахувати на ранніх стадіях розробки. На базовому рівні розраховується трудомісткість і вартість розробки як функція від розміру програми. Розмір виражається в оціночних тисячах рядків коду (KLOC - *kilo lines of code)*. Середній рівень COCOMO враховує безліч факторів, тоді як детальний рівень додатково враховує вплив окремих фаз проекту на його загальну вартість. На середньому рівні розраховується трудомісткість розробки як функція від разм єра програми і безлічі «чинників вартості», що включають суб'єктивні оцінки характеристик продукту, проекту, персоналу і апаратного забезпечення [67, 82].

До переваг методики можна віднести її орієнтацію на розробку ПЗ і роботу саме з тими потрібним і параметрами розробки, які використовуються при плануванні, методика дозволяє виробляти перерозподіл ресурсів,

наприклад, з метою зменшення термінів, як це було описано п. 1.3. Головним недоліком методики, як це вказую т самі автори, є її орієнтація на кількість рядків вихідного коду програми. А це далеко не завжди є критерієм трудомісткості. Другим недоліком є висока складність розрахунків і висока роль експертних оцінок при використанні середнього і детального рівня моделі.

1.4.5 Необхідність створення нового методу планування

Незалежно від обраного підходу, точний розрахунок ресурсів, необхідних для реалізації даного продукту за певних вимог до якості, є однією з основних п роблю в галузі управління проектами. Виникає реальна потреба в розробці методів і засобів, що дозволяють керівнику оцінити необхідні тимчасові і людські ресурси на основі всіх наявних характеристик проекту: історії попередніх подібних ін оектов, досвіду та продуктивності співробітників, специфіки компанії і т.п. Крім того, потрібно також мати можливість перерахунку і уточнення термінів і ресурсів вже на етапі розробки системи з урахуванням поточних тенденцій, які спостерігаються при реалізації проекту . Це допоможе керівнику своєчасно виявити відхилення від встановленого графіка і вжити відповідних заходів в управлінні проектом.

1.5 Огляд і аналіз систем ведення проекту

Інтенсивне розвиток ін формаційних і комунікаційних технологій в останні роки призвело до формування нових вимог до планування та управління проектами в різних областях. Одним з обов'язкових умов ефективної реалізації проектів стає застосування сучасних засобів та інструментів управління проектами, заснованих на використанні нових інформаційних технологій. Розвиток спеціального програмного забезпечення для планування та управління проектами обумовлено, в першу чергу, необхідністю максимальної інте грації найбільш ефективних методів, засобів і інструментів теорії управління проектами.

Діяльність практично будь-якого сучасного проекту, незалежно від тривалості, сфери діяльності, кількості і складу учасників, бюджету, намічених цілей неможливо уявити без використання сучасного програмного забезпечення.

Нижче розглянуто класифікацію та функціональні особливості сучасних систем планування і управління проектами, особливості їх практичного застосування, можливості інтеграції і сумісності з іншими додатками, програмами та системами, підтримка сучасних стандартів і принципів управління проектами.

В даний час на ринку представлено значну кількість універсальних програмних пакетів для персональних комп'ютерів, які автоматизують їз

функції планування і контролю календарного графіка виконання робіт. Найбільш популярними є такі:

- **Primavera Project Planner (P 3)** (Primavera);
- **Microsoft Project** (Microsoft);
- **Time Line** (Time Line Solutions);
- **Open Plan** (Welcome Software);
- **Artemis Views** (Artemis Management Systems);
- **CA - Super Project** (Computer Associates International Inc.);
- **Project Scheduler (Scitor Corp.);**
- **TurboProject (IMSI);**
- **Project Workbench (Applied Business Technology);**
- **Spider Project (Технології управління Спайдер);**

Багато фахівців з розробки та впровадження систем управління проектами поділяють ПЗ на професійні та настільні (непрофесійні). Професійні системи надано ють більш гнучкі засоби реалізації функцій планування і контролю, але вимагають великих витрат часу на підготовку і аналіз даних і високої кваліфікації користувачів. Другий тип пакетів призначений для тих, для кого управління проектами не є ос новной діяльністю. Від таких програм не потрібні будь-які специфічні функції планування або оптимізації розкладів. Для них більш важливим є простота використання і швидкість отримання результату [8, 52-54].

У джерелах зустрічаються і други е класифікації засобів управління проектами [14].

• засоби планування і контролю за виконання проектів;

• інструменти фінансового планування проектів;

• засоби створення і управління віртуальним офісом проекту. Однією систем ведення проектів є Open Plan Professional (рисунок 1.4). Ця система виділяється потужними засобами ресурсного планування, які дозволяють управляти різними видами ресурсів: людьми, обладнанням, матеріалами, фінансами, враховуючи їх взаємозв'язку. В Open Plan ін едусмотрени спеціальні процедури планування та контролю витрат, серед яких особливо варто відзначити засоби аналізу і побудови звітів по фактичний виробіток. Засоби «запам'ятовування» кількох варіантів реалізації проекту і введення фактичних даних п про витратам на роботу і відпрацювання ресурсів дозволяють проводити аналіз витрат за проектом, як прогнозованих, так і реальних. В Open Plan присутній аналітичний інструмент, що базуються на методі Монте-Карло, що дозволяє визначити можливі допуски в оцінці терміну завершення окремих робіт, цілих етапів і всього проекту, при цьому суттєвим

обмеженням є те, що необхідна для роботи методу імовірнісна оцінка робіт проводиться вручну.

Open Plan надає множес тво коштів, призначених для структуризації і візуалізації моделі проекту: ієрархічна структура завдань (WBS - Work Breakdown Structure), мережева модель (PERT-діаграма) і ін.

В Open Plan є можливість використання імовірнісних оцінок, отриманих в результаті аналізу ризиків проекту. Також підтримується автоматичне розпізнавання критичного шляху проекту, по введених даних про тривалості завдань і їх оцінками. Основною перевагою Open Plan є розгалужена система контролю над ресурсами і фінансами. Підтримуються безліч видів звітів по використанню різних видів ресурсів, а також великі можливості фінансів звітності проектів [52].

Одним з найбільш поширених програмних продуктів, призначених для управління проектами , є Microsoft Project . На сьогоднішній день остання версія даної програми - MS Project 2017 представляє собою широкий набір програмних інструментів для управління проектами, але, як відзначають багато користувачів MS Project , головним його плюсом крім наявних функцій є глибока інтеграція з іншими засобами і продуктами Microsoft і стандартний інтерфейс, звичний користувачам за іншими програмами з пакета MS Office [51].

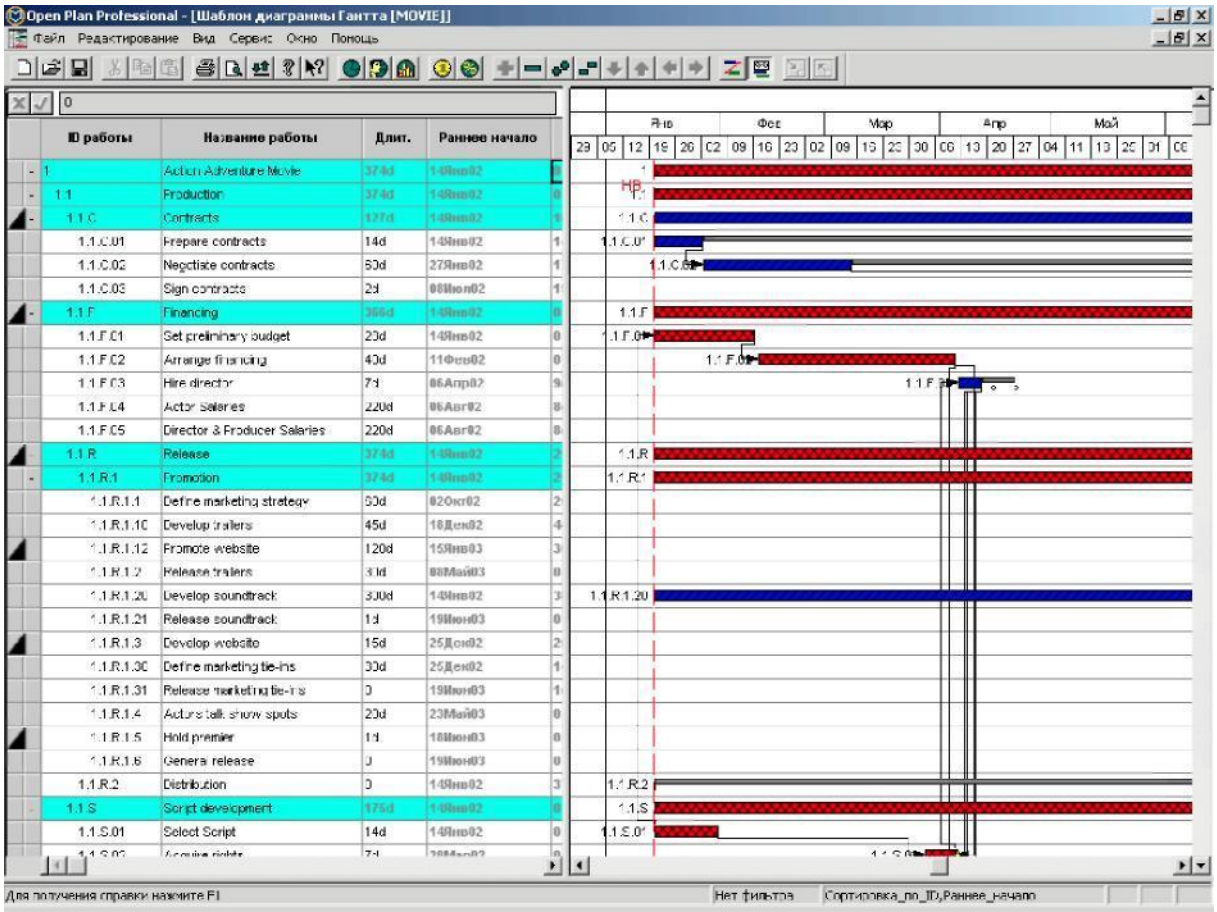


Рис. 1.4 - Управління ресурсами в Open Plan Professional

Microsoft Project дозволяє керівникам проектів управляти календарними планами і ресурсами, отримувати відомості про стан проекту і аналізувати його виконання. Застосовувані візуальні засоби планування дозволяють керівникам проектів швидко оцінити вплив змін, внесених в завдання або ресурси. Наприклад, якщо завдання відстає від календарного плану, M S Project показує вплив цієї зміни на інші завдання і на проект в цілому. Такі візуальні представлення даних проекту, як діаграма Ганта, мережевий графік і інші, дозволяють правильно оцінити залежності завдань і стан проекту.

MS Project позиціонується як система для керівників проектів, як для відстеження завдань, т ак і для контролю витрат і ресурсів та управління ними. Для аналізу проектів в MS Project є підтримка методу PERT [49].

Особливий інтерес представляє російська розробка - Spider Project (рисунок 1.5) [48]. Розробники Spider Proj ect - пакета управління проектами, заявляють такі відмінні риси:

• автоматична компоновка розкладу виконання робіт і оптимальне використання ресурсів проекту;

• можливості з обліку продуктивності призначених ресурсів, облік кваліфікації ресурсів;

• вбудована система аналізу ризиків та управління резервами за термінами і вартістю робіт;

• можливості по розрахунку ймовірностей успіху на основі розширень методу PERT ;

• широкі можливості вартісного і ресурсного аналізу проект ів. В цілому,

Sprider Project може бути розглянутий як гідна
альтернатива MS Project , з практично такими ж, а місцями і більш широкими функціональними можливостями, поступаючись останньому в основному за рахунок взаємної інтеграції з іншими додатками MS [50].

Серед дорогих професійних рішень виділяється система управління проектами Primavera Project Planner (P3). Система застосовується для великих проектів, дозволяє вирішувати складні ресурсні конфлікти, має розширені засоби анал іза використання ресурсів. Ще однією особливістю є підтримка різних рівнів доступу і різних автоматизованих робочих місць. Дозволяє проводити введення і оновлення даних безпосередньо з PERT -уявлення, календарно-мережного плану і брешемо енной логічної діаграми. Також має функції аналізу проекту за допомогою методу PERT.

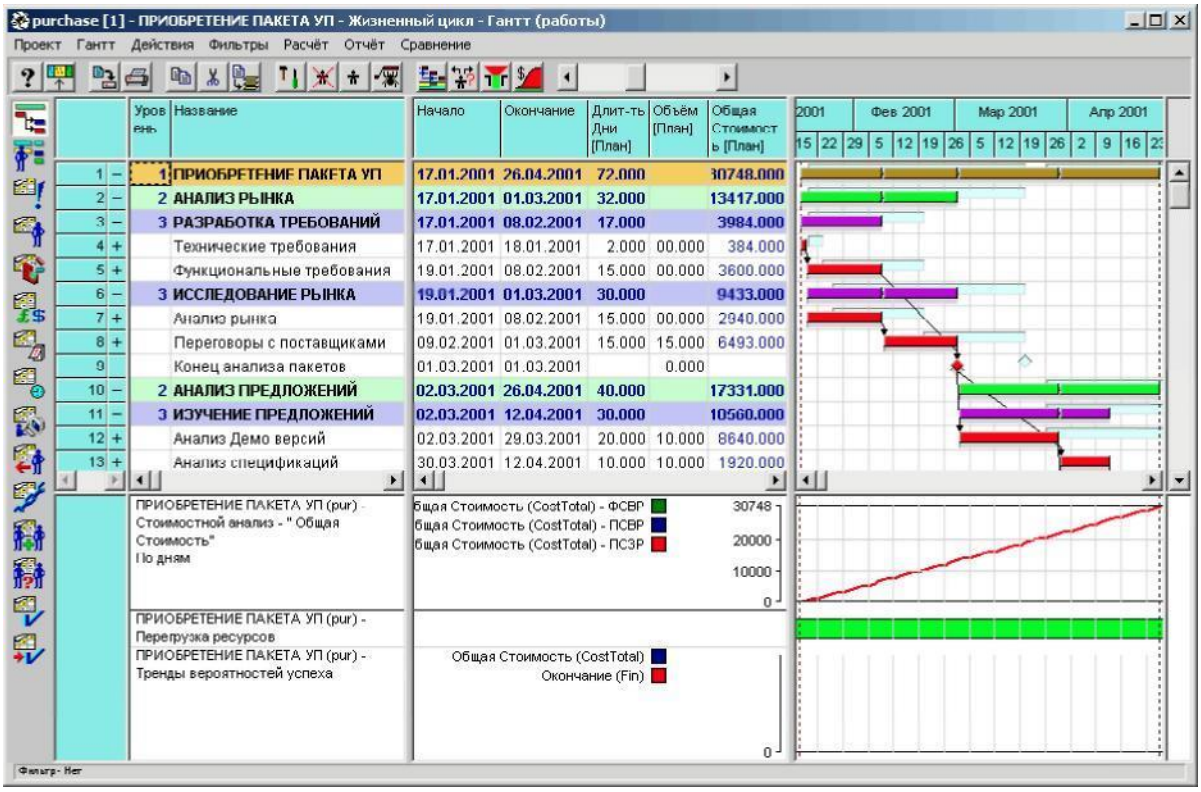


Рис. 1.5 - Діаграма Ганта життєвого циклу в Spider Project

Розроблена для великих проектів. У системі є підтримка проектів містять до 100,000 робіт, PERT -представлення спрощує завдання логічних послідовностей робіт. При плануванні проекту можливо відстежувати і аналізувати резерви, залежності між роботами і вузькі місця проекту. Р 3 дозволяє контролювати групи проектів, координуючи незалежні проекти в многопроектной оточенні і з огляду на залежності між роботами різних проектів. Для роботи з ризиками або при виявленні проблемної ділянки Р 3 дозволяє перебрати десятки можливих шляхів для найкращого рішення і найбільш ефективного використання критичних ресурсів. За допомогою набору засобів Р 3 легко простежити (як детально, так і укрупнено на різних рівнях структури проекту) результат різних впливів на проект, які можна зробити в ситуації, що склалася.

Наступну групу систем складають рішення, спрямовані переважно на планування і управління фінансовою стороною проекту. У них використовуються в основному прямі залежності складових проекту.

Так, система управління бюджетом проектів - «Cobra» призначена для аналізу і контролю фактичних даних по виконанню бюджету і прогнозування бюджету на основі поточних витрат і вартості ресурсів. Система Project Expert в доповненні до аналогічного функціоналу має великі можливості по моделюванню діяльності підприємства, і за своїм складом властивостей може бути віднесена більше до case - засобам, ніж до систем управління проектами.

Таким чином, сучасні системи календарного планування надають основний набір функціональних можливостей:

1. Кошти, призначені для опису робіт проекту, зв'язків між роботами і їх тимчасових характеристик, побудова мережевого графіка проекту. До них відносяться підтримка калі ндаря проекту, обмеження, що накладаються на роботи проекту, можливості призначення тимчасових характеристик і обмежень, зв'язку між завданнями і допустимі витрати.

2. Кошти підтримки інформації про ресурси та витратах за проектом і призначення ресурсів і витрат окремих робіт проекту. До них відносяться різні інструменти, призначені для ресурсного планування і оптимізації навантаження на ресурси проекту [30].

3. Засоби контролю над ходом виконання проекту та використанням ресурсів, включающі е інструменти відстеження стану завдань проекту, відсоток завершення.

4. Зручні графічні засоби подання структури, візуального побудови діаграм, засоби складання звітів.

1.6 Висновки

1. Проаналізовано різні визначення проекту та дано визначення проекту та проектної діяльності з точки зору розробки ПЗ.

2. Виявлено основні чинники, що перешкоджають точному розрахунку часу закінчення проекту.

3. Проведено аналіз обмежень процесу розробки програмного забезпечення. В ході нього були проаналізовані і класифіковані основні підходи до розробки ПЗ; виділені їх особливості, які необхідно враховувати при створенні методів планування.

4. Були проаналізовані відомі методи розрахунку термінів виконання проектів, визначено їх переваги та недоліки. Аналіз ходу реальних проектів показав, що більшість з методів не надають користувачам необхідну точність при плануванні проектів розробки програмного забезпечення. Серед причин цього можна виділити в першу чергу їх націлену дію «в цілому», без зазначення конкретної області застосування даної методики. Тому розробка методів, що дозволяють більш точно планувати час закінчення IT-проекту є актуальним завданням.

РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОЇ СТРУКТУРИ ПРОЦЕСУ ВИКОНАННЯ ПРОЕКТУ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Загальна схема і етапи виконання проекту

У першому розділі були розглянуті різні методики ведення проекту, але в загальному випадку вони відрізняються, головним чином, принципами організації самого процесу розробки програмного забезпечення, але в будь-якому випадку, в тому чи іншому вигляді присутні наступні етапи (рисунок 2.1):

- початок проекту (ініціація);
- формування вимог;
- формування списку завдань;
- формування команди;
- складання плану завдань;
- виконання завдань;
- аналіз результатів проекту.

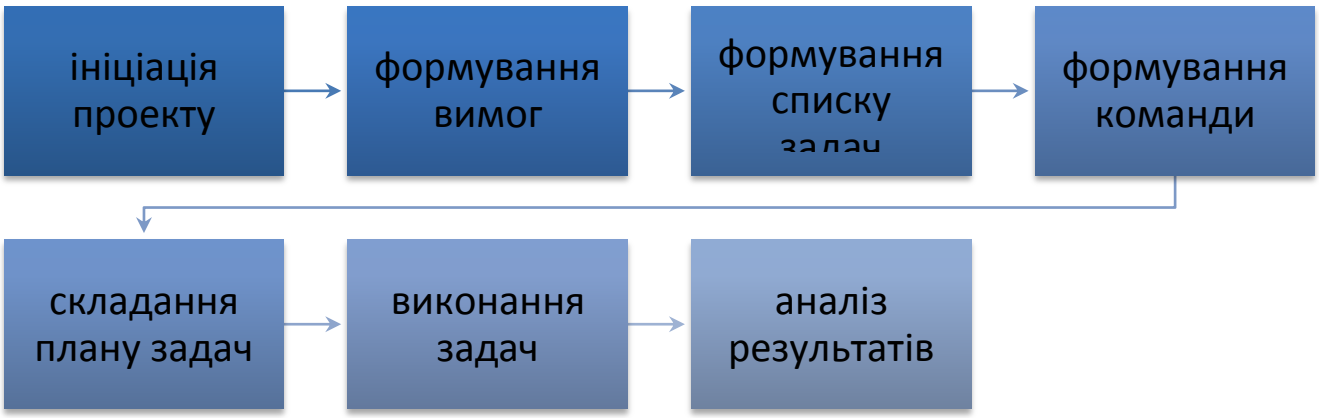


Рис. 2.1 - Типова схема виконання проекту

Під виконанням завдань мається на увазі вся діяльність по програмуванню, тестування, розробці дизайну і т.д. Аналіз підходів саме до побудови цього процесу, описаних в різних методологіях, був проведений в розділі 1 і виявив переваги методології Agile [29].

У загальному випадку, незалежно від використовуваної методології, такий підхід до ведення проекту має великий недолік - проаналізувати і спробувати усунути помилки планування можна тільки після завершення проекту або ітерації, а це часто коштує великих затрат часу і ресурсів.

Тому перед початком виконання завдань з розробки, а також після виконання чергової ітерації, необхідно мати можливість отримати прогноз ходу і результатів виконання і термінів закінчення проекту для поточного плану проекту, і внести коригування, якщо є відповідна потреба.

Аналіз існуючих систем ведення проекту показав, що вони все надають користувачеві широкий спектр можливостей по складанню плану проекту та фіксації процесу його виконання, інтеграції з системами контролю версій і відстеження помилок і т.д., але не володіють достатніми можливостями прогнозування ходу проекту і термінів його завершення.

У деяких з них, наприклад, в Microsoft Project , вбудована підтрика методу PERT , що дозволяє визначити приблизний термін виконання проекту і ймовірність виконати його рано чи пізно терміну, в цьому випадку схема ведення проекту має вигляд, представлений на малюнку 2.2.

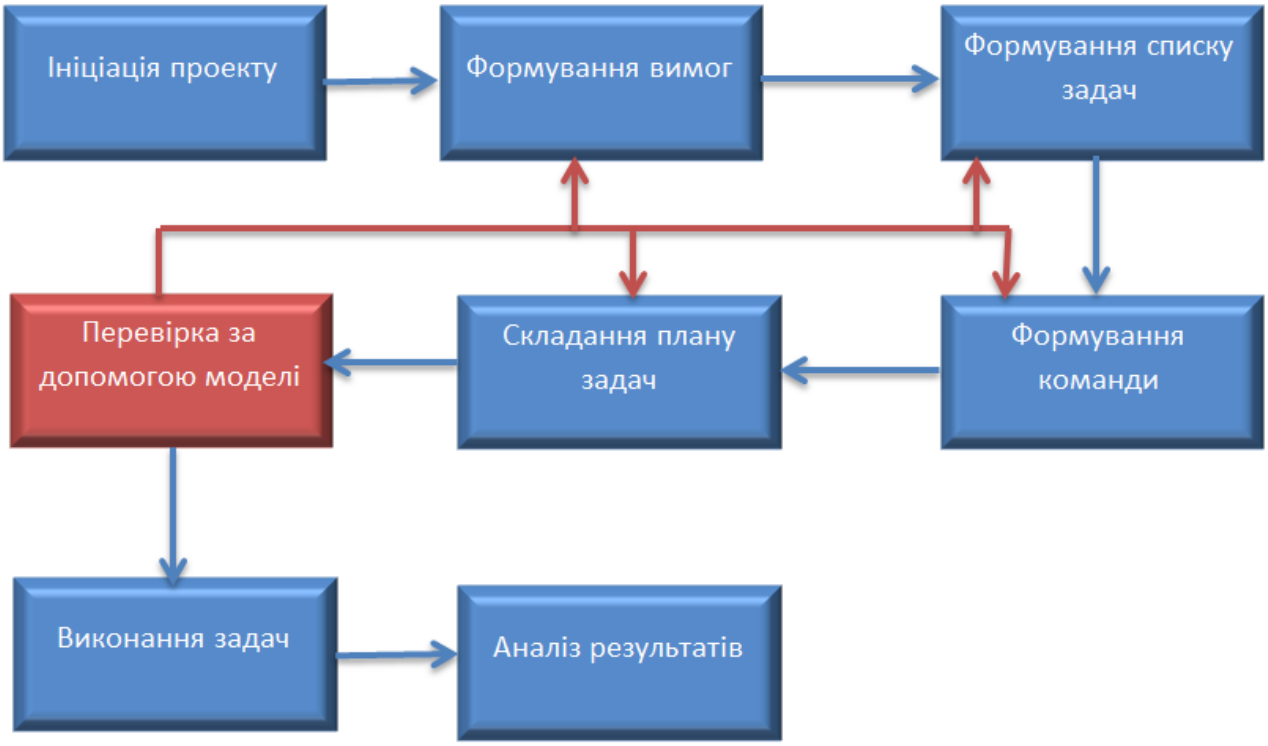


Рис. 2.2 - Схема виконання проекту з урахуванням коригування по результатами аналізу

Але точність такої оцінки дуже низька у зв'язку зі складністю точного завдання залежностей завдань в проектах розробки ПЗ, необхідних для побудови мережевого графіка всього проекту, недостатньою початкової точністю оцінок, одержуваних з використанням технології PERT , а також проблематичністю внесення змін до нього в ході проекту.

Методики COCOMO / COCOMO II і програми, які їх реалізують, також позволя ють проводити попередні розрахунки проекту за допомогою деякої моделі, але вони дуже неточні на базовому рівні розрахунку і дуже складні, фактично представляючи собою окремий проект, при використанні середнього або детального рівня, при це цілий ряд пара метрів також оцінюється суб'єктивно [78-79].

Існуючі системи ведення проектів (MS Project , Primavera Project Planner , Spider Project і т.д.) надають кошти для опису робіт проекту, зв'язків між роботами і їх тимчасових характеристик, постро ення мережевого графіка проекту, підтримки інформації про ресурси та витратах по проекту і призначення ресурсів і витрат окремих робіт проекту, контролю над ходом виконання проекту та використанням ресурсів, що включають інструменти відстеження стану за дач проекту, відсоток їх завершення, зручні графічні засоби представлення структури, візуального побудови діаграм, засоби складання звітів, але не дозволяють в повній мірі мати уявлення про подальший перебіг процесу виконання проекту, це св язано зі складністю інформаційних структур, з якими необхідно працювати при прогнозуванні процесу розробки ПЗ .

Для створення інструментальних засобів підтримки прийняття рішень при управлінні проектом розробки ПЗ з можливістю прогнозування хо да його виконання необхідно детально досліджувати інформаційні процеси і структури даних, які в ньому задіяні.

2.2 Функціонально-інформаційна модель процесу виконання IT-проекту

2 .2 .1 Ініціація проекту і складання списку завдань.

На різних етапах IT-проекту необхідно обробляти різну за своїм складом, ступеня впорядкованості і структурі інформацію. У залежимо ості від виду інформації, з якою доводиться мати справа етапи проекту були об'єднані в три фази: ініціація проекту і складання списку завдань, побудова плану ітерацій проекту, виконання завдання по п роект. Після завершення всіх трьох фаз процесу зазвичай проводиться аналіз результатів. Схема процесу виконання IT-проекту представлена на IDEF 0-діаграмі (рисунок 1) в додатку 2.

Перша фаза включає в себе наступні процеси:

• Процес збору вимог і формування набору завдань за проектом

Після формування мети проекту та визначення загальних вимог робота над проектом починається з уточнення вимог та постановки завдань. Основною складністю на цьому етапі є різнорідність і недолік в чная ступінь формалізації вимог до розроблюваних систем. Крім того, вимоги можуть виходити з різних

джерел - осіб або організацій, зацікавлених у виконанні проекту. Ними можуть бути замовник (і) проекту, аналітики, які вивчають ринок, станда ртізірующіе організації, юридичний відділ і т.д.

Як джерело нових вимог до системи можна також розглядати помилки, які виявляються на етапі тестування, оскільки в цьому випадку завдання або її частини повинні бути виконані заново. Від кожного і сточнік з різною інтенсивністю надходять різнорідні, що володіють різними пріоритетами, вимоги до розроблюваної системі. Підсумковий пріоритет вимоги при цьому становить суперпозиція пріоритету джерела і пріоритету цієї вимоги в рамках іст очники. У найзагальнішому випадку необхідно розглядати нескінченну кількість вимог.

Від кожного джерела з різною інтенсивністю надходять різнорідні, що володіють різними пріоритетами, вимоги до розроблюваної системі. У найзагальнішому випадку необ ходимо розглядати нескінченну кількість вимог. На основі всіх вимог складається попередній загальний список завдань. Схема процесів першої фази проекту представлена на малюнку 2 в додатку 2. • Процес побудови графа завдань.

При плануванні проекту зазвичай на основі попереднього загального списку завдань з урахуванням їх зв'язків і пріоритетів керівник проекту становить граф завдань. Список завдань, отриманих від джерел завдань, аналізується і перетворюється до виду графа, який відповідає списку завдань по одному проекту з урахуванням пов'язаності завдань, подзадач і їх пріоритетів.

Цей процес досить трудомісткий навіть для проектів середнього розміру кількість завдань, які беруть участь в оцінці може бути більше сотні; визначивши їх взаємозв'язку, часто буває досить важко розрахувати критичний шлях вручну, автоматизація цього процесу дозволяє прискорити аналіз план-графіка проекту, а також дає можливість моделювати можливі варіанти ходу течії проекту за методом Монте Карло. • Процес формування загального списку завдань

Після побудови графа завдань необхідно проаналізувати його з точки зору узгодженості завдань по їх зв'язках і пріоритетам. Так, наприклад, одна з підзадач може мати більш високий пріоритет, ніж батьківська чи попередні їй, в такому випадку пріоритет завдань необхідно переглянути.

Для формування черги завдань проекту за складеним графу необхідно оцінити час виконання (трудомісткість) кожного завдання. Існують різні методики оцінки завдань, для прогнозування проектів найважливішим моментом є не абсолютна величина оцінки, а відносні оцінки завдань, порівнюючи час виконання завдань між собою може бути уточнений статичний коефіцієнт для визначення похибки ості оцінок.

Наступний крок - декомпозиція їх на більш дрібні, зручні для вирішення завдання. Рівень декомпозиції залежить від багатьох факторів, але на практиці зручно вибирати такий розмір завдання, яка була б закінченою і могла б бути інтегрована в розроб бативает ПЗ . Для подальшої оцінки проекту цілком дуже важливо, щоб такі завдання виконувалися за приблизно однакові інтервали часу, т. Е. Для різних проектних завдань рівень декомпозиції був однаковим з точки зору часу їх виконання.

Після проходження оцінки і обробки виходить розмічений за вагами і упорядкований за пріоритетами граф завдань проекту, на основі рекурсивного обходу графа завдань відповідно до пріоритетів створюється загальний сп Ісок завдань проекту (англ. " Backlog ") зі збереженням інформації про пріоритети і взаємозв'язках.

2.2.2 Побудова плану ітерацій проекту

Головним завданням другої фази є отримання плану проекту та формування прогнозу його виконання. До завдань цієї фази так само входить оцінка ймовірності виконання плану проекту за допомогою однієї з моделей і перестроювання отриманого плану в разі незадовільного прогнозу.

До основних процесів другої фази відносяться:

• Складання плану проекту

Для складання плану проекту попередньо необхідно вибрати довжину окремих ітерацій проекту, при проходженні Водоспадної моделі будь- проект складається тільки з однієї ітерації. Як правило, ітерації вибираються одінаковог про розміру за часом, навіть якщо команда розробки не дотримується гнучких методологій розробки.

У простому випадку, коли н е потрібно враховувати залежності між проектами, є стабільна команда розробки та ін., План ітерацій проекту можна отримати, розподіляючи завдання із загального списку завдань проекту між членами команди розробки, поки не закінчиться доступне час ітерації. Аналогічно процес повторюється для інших ітерацій, поки не будуть розподілені всі завдання зі списку. В результаті вийде розрахункова дата закінчення проекту і кількість планових ітерацій проекту.

• Оцінка пл ана проекту за допомогою однієї з моделей

Цей етап є одним з найважливіших, проте в багатьох проектах він пропускається. Терміни, отримані на попередньому етапі, є грубим наближенням. Для того, щоб обчислити дату закінчення проекту з більш високо й точністю застосовується будь-яка (або навіть їх поєднання) з методик попередньої оцінки проекту. Ручне виконання цього завдання - дуже трудомісткий процес, наприклад, для методу Монте-Карло, ручні розрахунки

імітаційної моделі і всіх її «результатів» буде потрібно вали б величезна кількість часу. Тому для розрахунків за допомогою моделей застосовуються відповідні програмні засоби.

• Перегляд плану проекту

Якщо в результаті оцінки за допомогою будь-якої моделі була отримана незадовільна ий прогноз успішного закінчення проекту, то план проекту переглядається і його побудова повторюється.

2.2.3 Процес виконання завдань

Під виконанням завдань по IT-проекту мається на увазі вся діяльність по програмуванню, тестуванню, розробки дизайну і т.д. Ця фаза зазвичай займає найбільше часу і на ній задіяна вся команда розробки. Загальна схема процесу виконання завдань представлена на малюнку 2.4. У разі зміни складу вхідних даних може виникати необхідність повернення до попередніх фаз.

Вимоги до ПЗ включають в себе бізнес-вимоги і системні вимоги. У первинному варіанті бізнес-вимоги, що надходять від різних джерел, зазвичай представлені у вигляді текстового опису, структурованого у вигляді списку, на основі аналізу якого з залишаються системні вимоги або технічного завдання, для опису яких використовується структурований текст або UML діаграми прецедентів в графічній або текстовій формі. Після цього вирішується завдання перетворення кожного вимоги в одну або н есколько завдань і розстановки їх пріоритетів. Одному вимогу може відповідати одна або кілька завдань. На даному етапі **завдання** включає в себе наступну інформацію:

• *номер* - унікальний номер завдання в списку завдань від джерела;

• *назва* - заго спритний завдання, коротко характеризує її суть;

• *опис* - детальний опис завдання;

• *початковий пріоритет* - пріоритет в списку завдань від джерела.

Для створення завдань на основі вимог необхідний їх ретельний аналіз, залучення різних фахівців на тов для визначення детальних вимог та обмежень, визначення пріоритетів, тому автоматизація перетворення вимог до списків завдань скрутна. Крім того, не всі вимоги в проекті є обов'язковими, тому що на їх реалізацію завжди на закладаються обмеження ресурсів: термінів і команди розробки, що також враховується при визначенні пріоритету завдань.

2.3.2 Об'єднання списків завдань

Після опису вимог і списків завдань від усіх джерел складається **попередній загальний список завдань** за проектом, найбільшу складність являє собою визначення пріоритетів завдань в ньому на основі раніше заданих початкових пріоритетів і пріоритету джерела.

На даному етапі відбувається перетворення N списків завдань до єдиного списку, в якому **завдання** включає в себе наступну інформацію:

• *номер* - унікальний номер завдання в загальному списку завдань

• *назва* - заголовок завдання , коротко характеризує її суть;

• *опис* - детальний опис завдання;

• *пріоритет* - пріоритет завдання в загальному списку завдань.

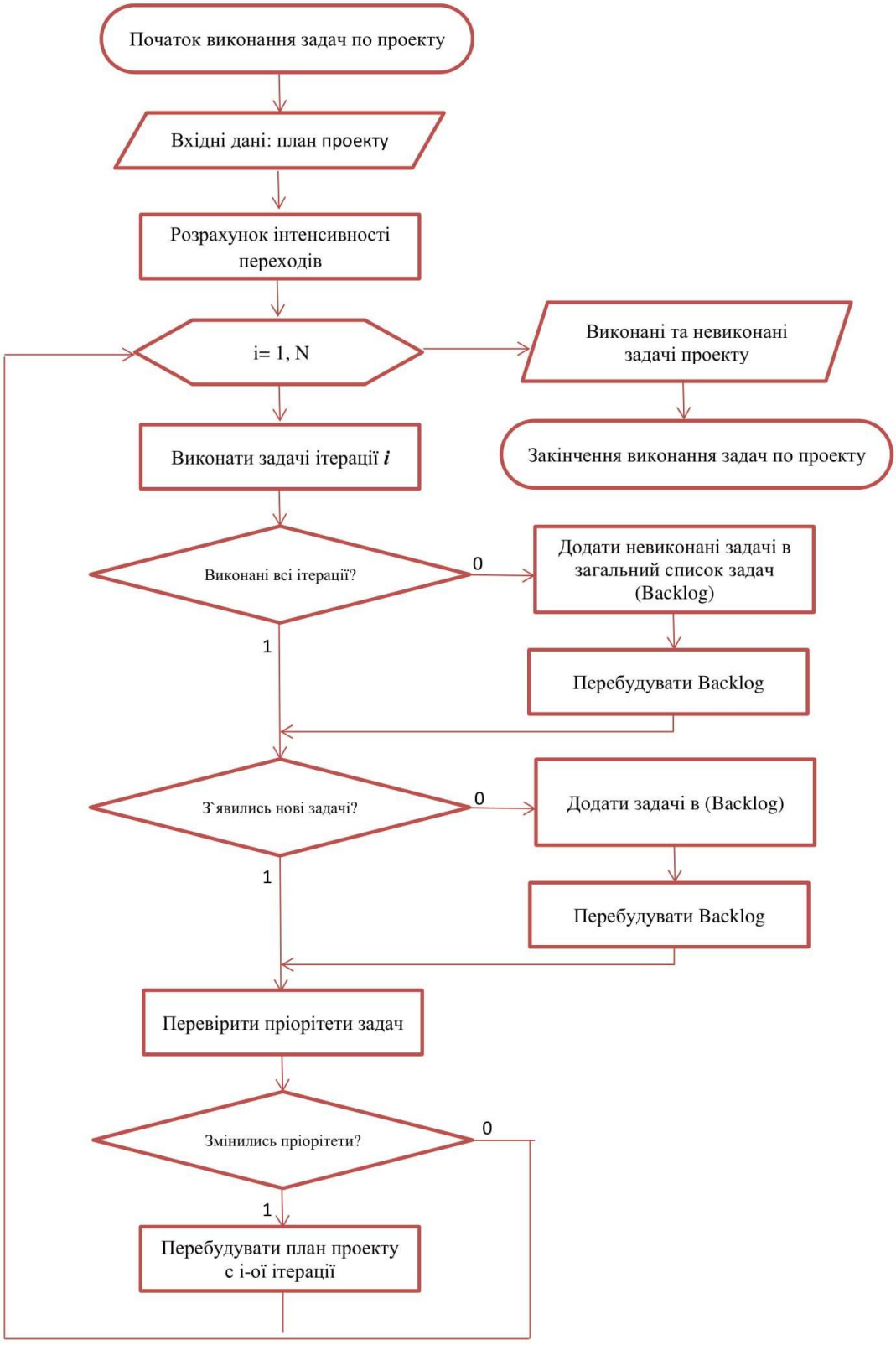


Рис. 2.4 - Схема процесу виконання завдань

2.3 Перетворення даних в процесі виконання IT-проекту

2 .3.1 Перетворення даних про вимоги до списків завдань від джерел

2.3.3 Побудова графа завдань

Задля побудови **попереднього загального списку завдань** за проектом для кожної задачі визначаються складність і її зв'язок з іншими завданнями. Види зв'язків можуть бути наступні:

1. Завдання А залежить від завдання В, тобто поки не виконана задача В, неможливо почати виконувати завдання А.

2. Завдання А є частиною завдання В, тобто поки не ви полнена завдання А, завдання В нічого очікувати виконано, при цьому трудомісткість завдання В дорівнює 0.

3. Завдання А асоціативно пов'язана з завданням В, тобто необхідні зміни функціоналу програмного продукту в завданні А, пов'язані з раніше зробленими змінами по завданню В.

Оцінка трудомісткості кожного завдання проводиться на основі експертного думки фахівців з урахуванням їх знань, досвіду минулих проектів і особливостей продукту, що розробляється.

Для спрощення оцінки трудомісткості завдання розбиваються на більш дрібні, в яких простіше оцінити обсяг необхідних змін. трудомісткість задач зазвичай вимірюється в людино-годинах, тобто кількості годин роботи одного фахівця.

Таким чином, на даному етапі опис завдання включає в себе:

• *номер* - унікальний номер завдання в загальному списку завдань;

• *назва* - заголовок завдання, коротко характеризує її суть;

• *опис* - детальний опис завдання;

• *пріоритет* - пріоритет завдання в загальному списку завдань;

• *трудомісткість* - оцінка складності завдання в умовних одиницях, наприклад, завдання в людино-годинах;

• *список залежних задач* - список номерів завдань, які залежать від поточної;

• *список підзадач* - список номерів завдань, котор ті є частинами поточної;

• *список пов'язаних завдань* - список асоціативно пов'язаних завдань з поточної.

З урахуванням інформації про залежності завдань список завдань перетвориться до **попередньою графу завдань** . Інформація про підзадача і ассоц іативно пов'язаних завданнях використовується виключно для спрощення роботи і контролю процесу їх виконання.

2.3.4 Перевизначення пріоритетів завдань

Побудова загального списку завдань для складання плану проекту ускладнюється на даному етапі неузгодженість ю побудованого графа завдань по пріоритетам (рисунок 2.6), тобто, наприклад, завдання Z 3 має перший пріоритет, але залежить від завдання Z 1 з третім пріоритетом, відповідно Z 3 не може бути виконана, поки не виконана Z 1, а в силу низького пріоритету Z 1 буде виконана не в перших ітераціях проекту. Для вирішення цієї проблеми перевизначаються пріоритети завдань, щоб привести залежні завдання до однаковим пріоритетам. Зазвичай підвищується пріоритет від завдань, які необхідні для вирішення інших високопріоритетних, тобто в прикладі підвищується до першого пріоритет завдання Z 1. Після цього **попередній граф завдань** перетвориться з **графу завдань** з узгодженими пріоритетами (Рисунок 2.7).

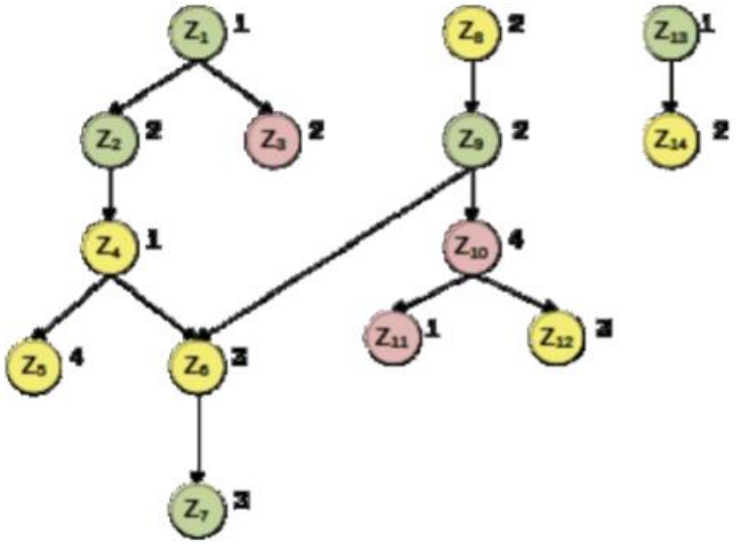


Рис. 2.6 - Граф завдань з неузгодженими пріоритетами

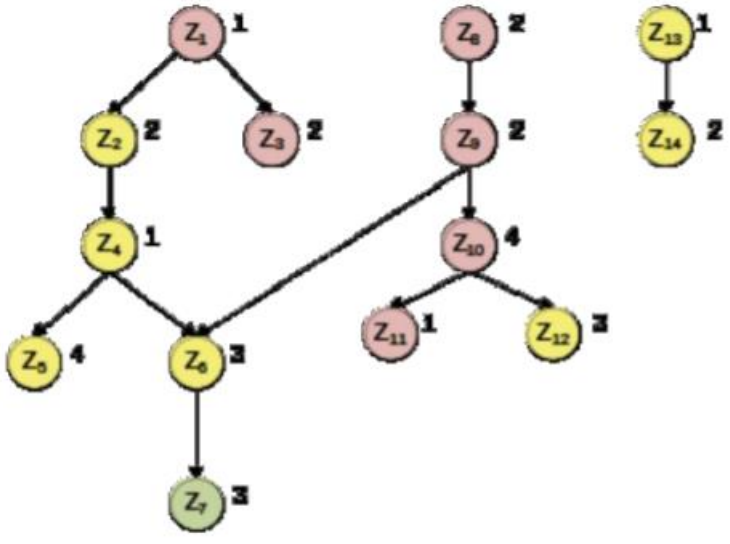


Рис. 2.7 - Граф завдань з узгодженими пріоритетами

У загальному випадку задача узгодження пріоритетів сказали ється шляхом їх обговорення керівником проекту з зацікавленими особами і командою розробки, при цьому пріоритети можуть бути перевизначені в бік зниження пріоритету залежною завдання. В описі завдання при цьому змінюються тільки пріоритети, але можу т додаватися нові завдання шляхом розбиття існуючих на більш дрібні з метою узгодити пріоритети, якщо для виконання залежною завдання досить вирішити частину попередньої. В тому випадку завдань і додаються в граф завдань.

2.3.5 Побудова загального списку завдань

На основі графа завдань будується загальний список завдань (" backlog "), опис завдання в якому включає в себе:

• *номер* - унікальний номер завдання в загальному списку завдань;

• *назва* - заголовок за дачі, коротко характеризує її суть;

• *опис* - детальний опис завдання;

• *пріоритет* - пріоритет завдання в загальному списку завдань;

• *трудомісткість* - оцінка складності завдання в умовних одиницях, наприклад, завдання в людино-годинах;

• *список залежних з Адачі* - список номерів завдань, які залежать від поточної;

• *список підзадач* - список номерів завдань, які є частинами поточної;

• *список пов'язаних завдань* - список асоціативно пов'язаних завдань з поточної;

• *статус* - стан завдання на даний момен т: нове завдання, в стадії розробки, виконана і т.д.

Д ля подальшого побудови **плану проекту** на даному етапі в опис завдань додається *статус,* щоб надалі враховувати зв'язку між завданнями. У кожній команді розробки може використовуватися свій набір статусів завдань в залежності від побудови процесу розробки і тестування.

2.3.6 Побудова плану проекту

План проекту включає в себе список ітерацій і план ітерацій. Залежно від списку завдань за проектом, його термінів і ви ділених ресурсів визначається довжина ітерації і їх кількість. Для побудови плану ітерацій завдання упорядковуються за пріоритетами, тобто будується чергу завдань з збереженням інформації про зв'язки між завданнями і їх статус.

На кожну ітерацію відбирається ряд завдань в залежності від їх пріоритету та трудомісткості, враховуючи статус завдань до даної ітерації для формування несуперечливого плану ітерацій з точки зору залежностей між завданнями.

План проекту є остаточними вихідними даними для виконання завдань по ньому. Після кожної ітерації він може змінюватися з урахуванням статусів завдань: якщо частина завдань в ітерації не було виконане, то це вимагає перегляду плану наступних за нею ітерацій з метою включения в них невиконаних завдань і перенесення залежних від них завдань.

У разі успішного проекту після завершення останньої запланованої ітерації всі завдання, які реалізують обов'язкові вимоги повинні бути виконані.

2.4 Автоматизація процесу розробки ПЗ

Проведений аналіз показав, що процес розробки ПЗ тривалий, трудомісткий, що характеризується великим числом перетворень даних на різних етапах проекту, особливо в питаннях, пов'язаних з роботою зі списками завдань за проектом в цілому або по ітераціям, тому що саме ці дані відображають всі внесені в план проекту зміни, як з боку зацікавлених осіб, так і сторони команди розробки. Для його спрощення застосовується ряд програмних засобів автоматизації на різних фазах створення проекту.

На початкових етапах першої фази, коли формуються вимоги і ще немає формалізованого списку завдань, автоматизацію процесу розробки становлять системи ведення проекту і системи зберігання вимог [91], що дозволяють вносити, зберігати, модифікувати, структурувати інформацію про вимоги, частина з них підтримує роботу з формалізованим поданням вимог у формі UML - діаграм прецедентів. Часто до цього додається можливість зберігання багатозв'язних списків вимог і завдань.

Повна автоматизація процесу формування та формалізації вимог і творення завдань на їх основі сильно утруднена занадто великим впливом всіх учасників проекту та інших зацікавлених осіб, від яких надходять вимоги або обмеження, тому для процесів першої фази використовуються тільки інструменти, що спрощують рутинну роботу з даними по проекту.

З підвищенням рівня формалізації інформації, що передається з одного етапу на інший у другій і третій фазах з одного боку по є більше повторюваної рутинної діяльності, яка може і повинна бути автоматизована, з іншого боку ще більше, ніж в першій фазі, виявляється динамічність процесу розробки ПЗ і мінливість вимог. Тому підвищується потреба у використанні засобів автоматизації та створення нових засобів підтримки прийняття рішень [90] для оперативного контролю за ходом виконання проекту.

В результаті проведеного аналізу інформаційної структури процесу розробки ПЗ були виявлені частини процес а, які потребують автоматизації:

• *перевизначення пріоритетів завдань і побудова загального списку завдань*

Складання загального списку завдань на основі попереднього графа відбувається в два етапи: перевизначення пріоритетів завдань і безпосередньо обхід гр афа для складання списку. Для вирішення обох завдань необхідно провести рекурсивний обхід графа, при кількості вершин перевищує десятки вручну ця процедура досить трудомістка, до того ж алгоритми рекурсивного обходу досить добре вивчені, поетом у автоматизація цього процесу повинна проводиться в першу чергу.

• *автоматизація побудови плану ітерацій і його перевірки з допомогою однієї або декількох моделей оцінки*

Як до початку виконання завдань, так і під час, план ітерацій може зазнавати множе ство змін, це пов'язано з мінливістю вимог до ПЗ і імовірнісним характером процесу їх виконання надалі, тому необхідно використовувати існуючі засоби моделювання для перевірки побудованого плану проек та з точки зору оцінки успішності виконання проекту і створювати нові для збільшення точності і спрощення отримання оцінок, що може досягатися шляхом автоматизації побудови та використання цих моделей.

• *автоматизований контроль ходу виконання завдань і внесення коригувань до плану проекту*

В ході виконання завдань велике значення має своєчасність реагування на зміни, для чого необхідно швидко отримувати актуальні дані по поточному стану завдань, аналізувати їх з метою визначення наслідків і вносити коригування до плану проекту. Для вирішення цієї проблеми необхідно створення нових засобів контролю ходу виконання проекту і систем підтримки прийняття рішень щодо внесення змін до плану ІТ-проекту.

Виходячи з цього, пропонується методика ведення проекту:

1. Перевірка плану проекту та його можливе коригування перед переходом до виконання завдань і після чергової ітерації.

2. Максимальний облік результатів попередніх проектів і завершених ітерацій при поточному плануванні.

3. Можливість простого внесення змін до плану проекту в процесі виконання завдань після чергової ітерації без зміни всього плану проекту цілком.

Схема процесу виконання завдань також буде включати в себе перебудову плану після завершення кожної ітерації з використанням моделі для попереднього розрахунку часу закінчення проекту (рисунок 2.10).

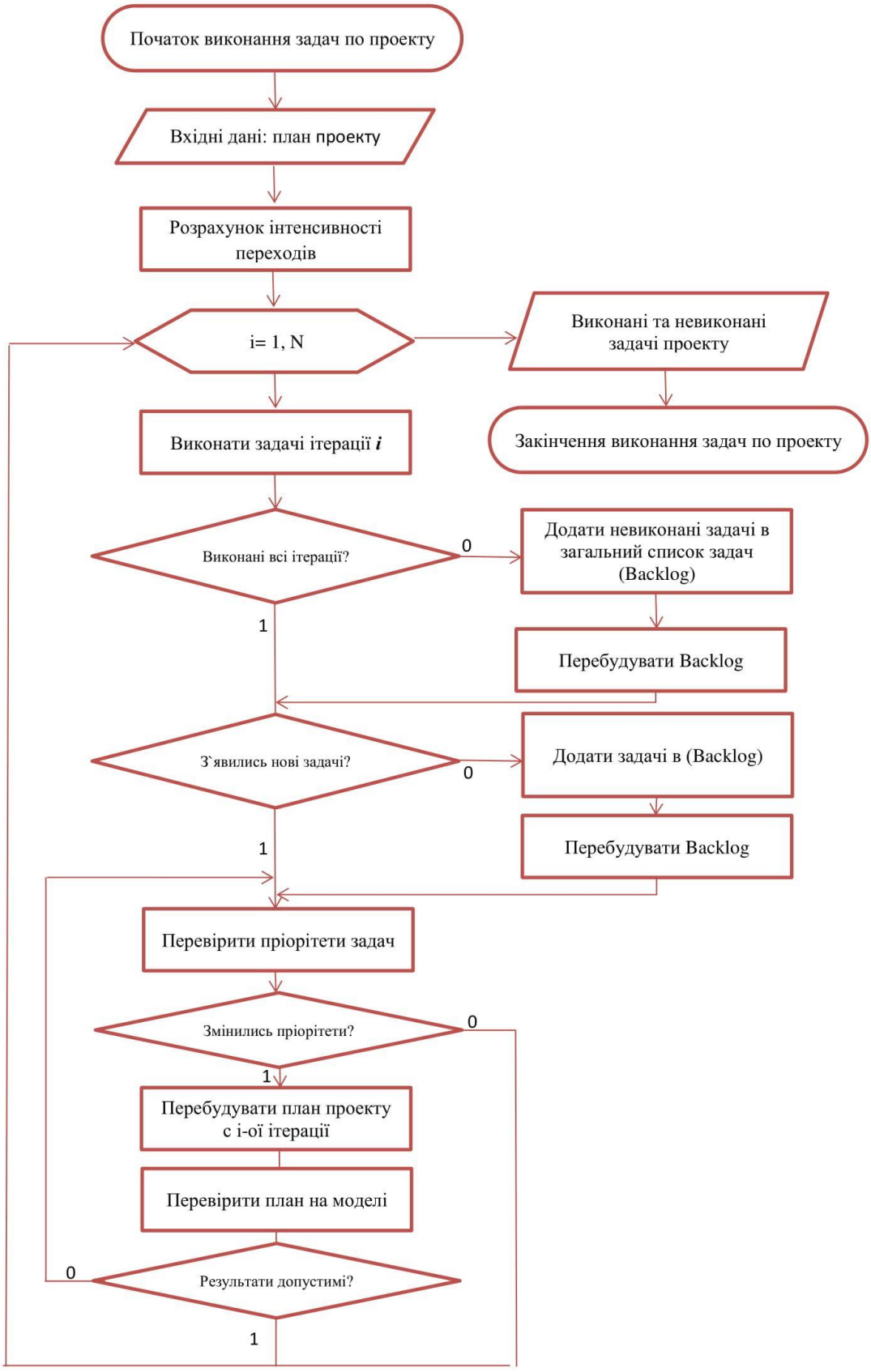


Рис. 2.10 - Схема виконання проекту з використанням перевірок на моделі плану проекту

Проведений аналіз інформаційної структури процесу розробки ПЗ показав, що в ньому задіяна велика кількість участни ков проекту і зацікавлених осіб, що впливають на хід його виконання на всіх етапах, при цьому крім безпосередньо виконання завдань щодо реалізації функціоналу ПЗ існує потреба в перетворенні великої кількості даних, які відображають динамічність процесу. Тому для застосування запропонованої методики необхідно створення інструментальних засобів підтримки прийняття рішень при плануванні проекту , які дозволять автоматизувати роботу з даними про завдання проекту на етапах планування і виконання завдань на додаток до існуючих засобів автоматизації управління вимогами та завданнями і на основі математічної моделі процесу розробки прорахувати основні характеристики майбутнього проекту, знайти їх прийнятне поєднання на основі наявних вхідних даних і тільки Після цього переходити до безпосередньо розробці.

2.5 Вимоги до системи підтримки прийняття рішень при плануванні і виконанні проекту

Для відображення особливостей процесу розробки ПЗ , автоматизації окремих його етапів і обліку специфіки широкого кола реальних IT-проектів система підтримки прийняття рішень повинна відповідати ряду вимог і обмежень. В першу чергу повинна надаватися можливість зміни різних параметрів, оскільки жоден проект не йде по заздалегідь наміченим планом і керівнику постійно доводиться адаптувати його до умов, що змінюються, а також можливість врахування особливостей команди розробки. Крім того, для забезпечення оперативної реакції на умови, що змінюються повинна бути можливість перерахунку плану проекту. Це дозволяє адаптувати план проекту до умов, що змінюються, враховувати особливості команди розробки. Крім того, така можливість перерахунку повинна бути присутня на кожному відносно невеликому відрізку проекту. Виходячи із сучасних тенденцій до зменшення і дроблення проектів, доцільно вибрати таким терміном один або кілька днів. Тобто повинна бути можливість щоденного коригування та внесення змін до списку характеристик проекту для побудови прогнозу. Детальна IDEF0 модель ведення проекту на основі запропонованої схеми представлена на рисунках .



Рис. - Блок A0 «Виконати IT-проект»

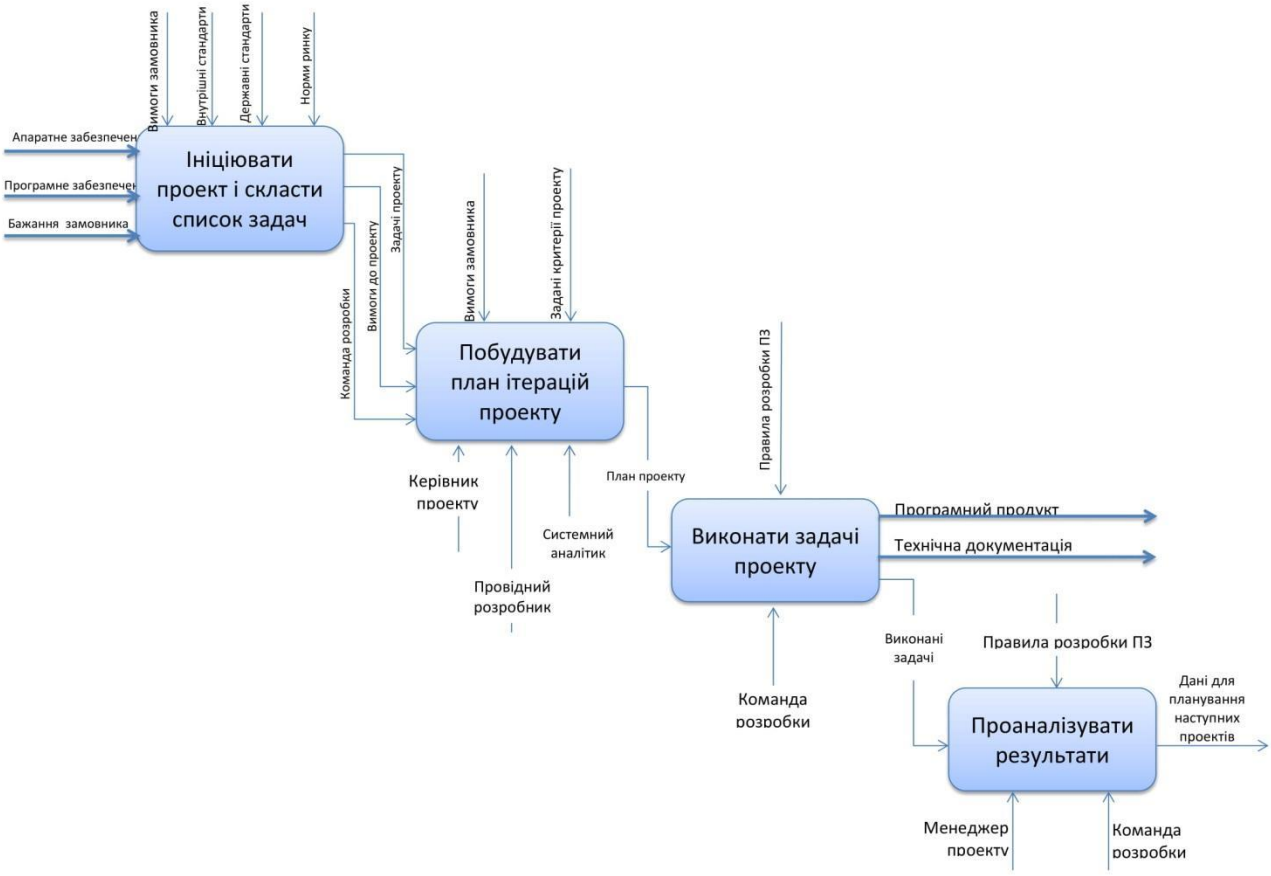


Рис. – - Декомпозиція блоку A0

Складові проект завдання можуть різнитися за важливістю для замовника проекту. Якщо команда розробки не виконує обов'язкові (пріоритетні) вимоги, то зазвичай такі проекти неуспішні, а невиконання навіть цілого ряду фонових вимог найчастіше ігнорується. На практиці також часто замовник проекту пред'являє до нього дуже велике число невеликих і переважно рекомендаційних вимог і висуває безліч пропозицій щодо можливих поліпшень. Такі вимоги, безумовно, повинні виконуватися після обов'язкових при наявності вільних ресурсів.

Найважливішим обставиною, яке повинна враховуватись ь система, є те, що завдання, час виконання якої оцінено в N годин, на самому ділі буде виконана за деякий випадковий час $N + AN$, а стверджувати, що завдання буде виконано рівно за відведений термін (або за будь-якої іншої) можна лише з некіт оройя ймовірністю. Тому і виконання всіх завдань ітерації або проекту цілком до певного терміну можна планувати тільки приблизно, домігшись необхідної ймовірності успішного завершення проекту. Система повинна дозволяти розраховувати цю вероятнос ть, надаючи можливість зміни основних параметрів проекту (час, кількість задіяних ресурсів, обсяг вирішуваних завдань).

При цьому практично завжди успішно завершеним вважається той проект, в якому виконані всі обов'язкові (необхідні д ля реалізації основної суті проекту) і будь-яку кількість необов'язкових вимог до заявленого терміну.

Для більшої наочності система повинна надавати різні варіанти розвитку проекту, особливо в разі незадовільного прогнозу: визначати оптимальну кількість завдань, прогнозувати час закінчення виконання сукупності всіх поставлених завдань, дозволяти вносити в ході виконання проекту, в залежності від його поточного стану, корективи в список завдань і склад команди розробки.

Таким чином, система підтримки прийняття рішень при управлінні проектамі розробки ПЗ повинна відповідати наступним вимогам:

1. Наявність коштів прогнозування ходу виконання і ймовірності успішного завершення проекту.

2. Наявність коштів визначення часу, необхідного для завершення виконання запланованих зад ач і визначення максимальної кількості завдань, які можуть бути виконані, в разі незадовільного прогнозу.

3. Можливість обліку результатів попередніх проектів і адаптації до поточного стану проекту.

4. Наявність засобів складання плану проекту.

5. Можливість зміни плану проекту в ході його виконання без повної перебудови процесу.

6. Можливість врахування пріоритетів завдань і зв'язків між ними.

7. Можливість обліку кваліфікації та індивідуальних характеристик членів команди.

2.6 Висновки

1. Виділено основні етапи виконання проекту, проаналізовані підходи до планування проекту без використання попередньої перевірки плану проекту та із застосуванням методу PERT і методики COCOMO .

2. Розроблено функціонально-інформацій а модель процесу виконання IT-проекту з розбивкою етапів проекту на фази з точки зору складу і структури оброблюваної інформації і виконуваних над нею операцій.

3. Проаналізовано процеси перетворення даних в ході виконання IT-проекту з точки зору можливості їх автоматизації.

4. Розроблено методику автоматизації процесу перетворення даних на етапах планування та виконання завдань з розробки ПЗ на додаток до існуючих засобів автоматизації управління вимогами та завданнями і розроблена схема виконання проекту з використанням системи підтримки прийняття рішень при плануванні проекту.

5. Розроблено алгоритмічне забезпечення та вимоги до системи підтримки прийняття рішень при плануванні проекту відповідно до запропонованої методикою.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Архітектура системи підтримки прийняття рішень при плануванні проекту

3.1.1 Основні модулі

Для розробки програмного засобу, що задовольняє виробленим вимогам, потрібно врахувати ряд особливостей IT-проектів , виходячи з цього система підтримки прийняття рішень при плануванні IT-проекту включає в себе (рисунок 3.1):

Модуль постановки завдань. Представлений інтерфейсом користувача для ведення проекту, створення і підтримки списку завдань, визначення їх трудомісткості, взаємозв'язків і пріоритетів, завдання команди розробки, її характеристик.

Модуль аналізу завдань. Включає в себе модуль побудови графа завдань і модуль створення черги. На основі заведених завдань за проектом, їх зв'язків і пріоритетів в модулі будується і аналізується граф завдань.

Після цього в модулі створення черги на основі побудованого графа завдань створюється черга завдань, яка згодом перетвориться до черги одиничних завдань.

Модуль прогнозування ітерацій проекту. Включає в себе модуль складання плану ітерацій, в якому на основі даних про проект і команду розробки, визначених у модулі постановки задачі, і отриманої черги одиничних завдань автоматично складається план ітерацій.

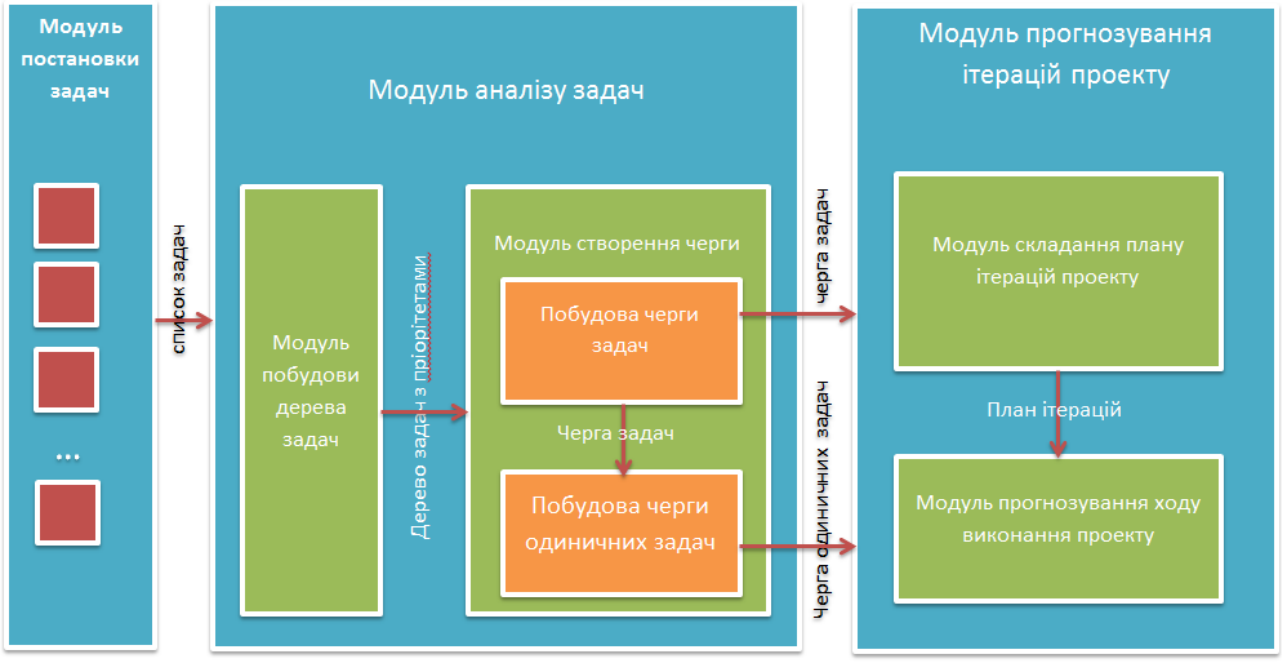


Рис. 3.1 - Архітектура системи підтримки прийняття рішень

На його основі в модулі прогнозування ходу виконання проекту складається прогноз успішності виконання поставлених завдань в задані терміни, в разі незадовільного прогнозу видаються рекомендації по оптимальній кількості завдань і часу виконання поставлених завдань.

Далі детально розглянуті основні модулі.

3.1.2 Модуль постановки завдань

Після визначення загальних вимог робота над проектом починається з уточнення вимог та постановки завдань.

Модуль постановки завдань надає користувачеві можливість ведення списку завдань по проектам і списку команд розробки. При додаванні нового проекту користувач задає його назву, характеристики, вимоги, терміни, основну документацію і переходить до додавання завдань за проектом.

На цьому етапі користувачів може бути кілька, тобто, як і в будь-якій системі управління проектами, вносити в список нові завдання можуть різні люди.

Для створення нового завдання необхідно, крім її назви, опису та вказівки, до якому проекту вона відноситься, задати її пріоритет і трудомісткість, вимірювану в одиничних завданнях. Тобто складність завдання визначається кількістю одиничних завдань, на які вона може бути умовно розбита.

Крім цього, користувач може вказати пов'язані між собою завдання, що виконуються послідовно, або підзадачі. Можливо також ведення списку команд розробки, завдання команди для кожного проекту, управління складом команди, створення нових команд.

При додаванні нового фахівця в команду необхідно вказати для нього інтенсивність виконання завдань. Ця характеристика може бути отримана після аналізу результатів попередніх проектів, в яких брав участь цей фахівець, або встановлена експертним шляхом. Згодом вона може коригуватися по результатам завершення чергової ітерації або проекту.

Згодом такі характеристики можуть коригуватися по результатами завершення чергової ітерації або проекту.

Таким чином, модуль постановки завдань забезпечує можливість додавання всієї необхідної інформації по проекту, включаючи документацію, задачі і команду розробки.

3.1.3 Модуль аналізу завдань

При плануванні проекту зазвичай на основі спільного з писку завдань з урахуванням їх зв'язків і пріоритетів керівник проекту становить план

виконання завдань. У розробленій системі підтримки прийняття рішень при управлінні проектами цей процес повинен автоматизований.

Список завдань, отриманих від джерел заду ч (замовників), аналізується і перетворюється до виду графа, який відповідає списку завдань по одному проекту з урахуванням пов'язаності завдань, підзадач і їх пріоритетів. Причому в системі передбачено використання пріоритетів двох видів: базовий пріоритет задачі і пріоритет джерела. Пріоритет джерела задається кожному користувачеві системи в залежності від його впливу на проект. Тому пріоритет завдання складається з цих двох характеристик, при цьому в першу чергу враховується базовий пріоритет завдання, а вже серед рівних за базовим пріоритетом завдань чергу будується з урахуванням пріоритету джерела. В результаті роботи модуля формується попередній граф завдань проекту.

Після побудови він аналізується з точки зору узгодженості завдань по їх зв'язках і пріоритетам. Так, наприклад, якщо одна з підзадач має більш високий пріоритет, ніж батьківська чи попередні їй, то їх пріоритет автоматично змінюється на більш високий пріоритет підзадачі. Результатом роботи модуля є сформований граф задачі.

Модуль створення черги завдань

Модуль створення черги призначений для оцінки вступників завдань, декомпозиції їх на більш дрібні зручні для вирішення завдання. В розробляється систем е декомпозиція проводиться до рівня одиничних задач, в яких вказується трудомісткість завдання при її створенні.

Після проходження етапу оцінки та обробки на вхід модуля створення черги завдань надходить розмічений за вагами і упорядкований за пріоритетами граф завдань проекту.

Даний модуль вирішує завдання рекурсивного обходу графа завдань відповідно до пріоритетів і зв'язками. На початок черги потрапляють завдання з більш високим базовим пріоритетом, що забезпечує їх першочергове виконання. Всередині списку задач з однаковим пріоритетом здійснюється сортування за пріоритетом джерела завдання. Залежно від зв'язків між завданнями для кожної з них встановлюється статус, який визначає чи доступна ця задача для вирішення в даний момент часу.

В результаті формується черга завдань за проектом з урахуванням всіх пріоритетів і взаємозв'язків (рисунок 3.2).



Рис. 3.2 - Черга завдань за проектом

Ще однією функцією даного м Одуля є перебудова наявної черги під час вступу нових завдань, як з боку замовників проекту, так і в результаті тестування проміжних версія ПЗ . Перебудова черзі відбувається за таким правилом: знаходиться останній елемент потрібного пріоритету, потім виконується вставка в уже наявну чергу, або в початок черги, якщо елементів з даними пріоритетом в черзі не знаходилося раніше.

Після створення черги завдань за проектом автоматично формується відповідна їй чергу одиничних завдань (рис. 3.3), тобто кожна задача замінюється на кілька одиничних завдань, еквівалентних їй по заданій джерелом трудомісткості, тобто задача з трудомісткістю ***M*** розпадається на ***M*** одиничних завдань, що йдуть підряд в черзі. Для кожної одиничної завдання з охороняється інформація про те, частиною якої задачі вона є.

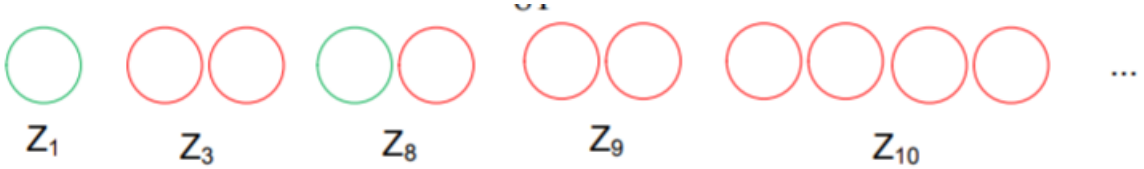


Рис. 3.3 - Черга одиничних завдань за проектом

При цьому керівник проекту може змінити т рудоемкость і порядок виконання завдань, скорегувавши відповідні пріоритети і характеристики трудомісткість.

3.1.4 Модуль прогнозування ітерацій проекту

Модуль складання плану ітерацій В результаті на вхід модуля прогнозування результатів виконання завдань надходить велика, в порівнянні з розміром одного етапу виконання проекту, чергу одиничних завдань. Далі в залежності від типу проекту може бути визначено:

• приблизну кількість ітерацій, необхідних для виконання *N* зад ач, при фіксованих рамках проекту. Ця кількість оцінюється як відношення *N* завдань до середнього кількості, яке колектив може виконати за одну ітерацію.

• приблизну кількість завдань, яке необхідно виконати за одну ітерацію при фіксовано м часу. Кількість завдань можна оцінити як відношення кількості завдань до кількості допустимих ітерацій проекту.

При цьому розмір ітерації відповідно до обраної ідеологією ведення проекту зазвичай фіксується для стандартизації процесів всередині ітерацій і. Однак в загальному випадку зустрічаються ітерації змінної довжини, тому система дозволяє враховувати і їх, оскільки за основу взята завдання одиничного обсягу.

Модуль прогнозування ходу виконання проекту

В основу роботи модуля прогнозування ходу виконання проекту покладена модель процесу виконання завдань з розробки ПЗ, в якій команда розробки розглядається як система масового обслуговування.

Вихідними даними для складання прогнозу є чергу одиничних завдань , план ітерацій для одиничних завдань і характеристики команди розробки: інтенсивність виконання завдань і кількість фахівців. План ітерацій включає в себе час і розмір кожної з ітерацій для підтримки роботи з нерівними ітераціями.

У модулі прогнозування результатів виконання завдань, що становлять проект, описана динамічна модель СМО використовується для отримання ймовірностей виконання списку запланованих на ітерацію завдань, відображення прогнозу ходу проекту, а також визначення необхідних для ітерації проекту ресурсів.

За допомогою отриманих залежностей визначається ймовірність виконання всіх запланованих завдань за ітерацію і в ероятності виконання кожного завдання по всіх днях в ітерації. Якщо ймовірність менше заданого в системі порогового значення, то визначається оптимальна кількість завдань для заданих вхідних параметрів і час закінчення проекту.

Після цього зберігається модель системи для поточних параметрів і виводяться результати прогнозу. Використовуючи побудовану модель, складається прогноз ходу виконання проекту протягом часу ітерації і визначається ймовірність успішного виконання всіх поставлених завдань за цей термін.

Задовільному прогнозу визначається порівнянням з граничним значенням ймовірності успішного виконання проекту, яке в системі задано за замовчуванням і так само 0.9 (90%).

Модуль прогнозування ходу виконання проекту може працювати в двох режимах: прогнозування і аналізу. У разі роботи в режимі прогнозування система будує прогноз (рисунок 3.5), який включає в себе набір одиничних за дач на ітерацію і кількість днів у відповідності до плану ітерацій, для вирішення кожного кількості завдань в кожен день ітерації розраховується ймовірність їх виконання, визначається ймовірність виконання всіх поставлених завдань за термін ітерації і оцінюється прийнятний результат прогнозу з урахуванням граничного значення.

№ дня / № задачі	1	2	3	4	…	K	План на ітерацію
1	p_{11}	p_{12}	p_{13}	p_{14}	…	p_{1k}	**p_1**
2	p_{21}	p_{22}	p_{23}	p_{24}	…	p_{2k}	**p_2**
3	p_{31}	p_{32}	p_{33}	p_{34}	…	p_{3k}	**p_3**
4	p_{41}	p_{42}	p_{43}	p_{44}	…	p_{4k}	**p_4**
…	…	…	…	…	…	…	**…**
d	**p_{d1}**	**p_{d2}**	**p_{d3}**	**p_{d4}**		**p_{d6}**	**p_d**
d+1	$p_{(d+1)1}$	$p_{(d+1)2}$	$p_{(d+1)3}$	$p_{(d+1)4}$	…	$p_{(d+1)k}$	**$p_{(d+1)}$**
d+2	$p_{(d+2)1}$	$p_{(d+2)2}$	$p_{(d+2)3}$	$p_{(d+2)4}$	…	$p_{(d+2)k}$	**$p_{(d+2)}$**
…	…	…	…	…	…	…	**…**
$d+d_{доп}$	**$p_{(d+dдоп)1}$**	**$p_{(d+dдоп)2}$**	**$p_{(d+dдоп)3}$**	**$p_{(d+dдоп)4}$**		**…**	**$p_{(d+dдоп)}$**

Рис. 3.5 - Результат роботи в режимі прогнозування

У разі роботи в режимі аналізу крім складання прогнозу після оцінки задовільному результату прогнозу система пропонує користувачу рішення по оптимізації плану ітерацій.

Якщо ймовірність успішного виконання завдань більше порогового значення, то система становить рас ширення прогноз для позитивного випадку (рисунок 3.6) і визначає кількість додаткових одиничних завдань, яке можна додати в аналізовану ітерацію, за умови, що ймовірність їх успішного виконання залишиться в допуску.

Якщо ймовірність успішного виконання завдань менше порогового значення, то система будує розширений прогноз для негативного випадку (рисунок 3.7) і пропонує змінити можливі параметри:

• збільшити розмір ітерації (кількість днів, необхідних на виконання безумовно го числа завдань);

• зменшити кількість виконуваних завдань, залишивши розмір ітерації незмінним. Така методика пропонується більшістю методологій, що відносяться до сімейства «гнучких»;

• змінити соста в команди розробки, додаючи в неї нових фахівців. При цьому в системі буде справить новий розрахунок середньої продуктивності команди і на її основі перебудується модель процесу виконання завдань і буде розрахована скоригована ймовірність виконан ня N відібраних завдань за час d .



Рис. 3.6 - Результат роботи в режимі аналізу для задовільного прогнозу

Оптимальна кількість завдань, яке можна з допустимою ймовірністю виконати за час ітерації і оптимальний час виконання поставлених завдань автоматично розраховуються системою. На основі них складається розширений варіант прогнозу.

Таким чином, в результаті роботи модуля прогнозування результатів виконання з Адачі користувачеві системи надається прогноз, складений на основі плану ітерацій, із зазначенням ймовірності виконання кожної з них в термін [39].

Для обліку того факту, що завдання не завжди відповідають своїй споконвічній оцінці, а також через появу будь-яких можливих додаткових деталей, які не обліковані при початкової оцінці, модуль прогнозування результатів виконання завдань показ ивает поточний статус ітерації, ймовірності виконання решти завдань до кінця ітерації, а також дозволяє зробити їх перерахунок при коригуванні кількості завдань, термінів або розміру команди, аналогічно тому, як це робилося перед початком ітерації.

На основі отриманих даних керівник проекту може прийняти отриманий план проекту і перейти до його реалізації або ж змінювати його до тих пір, поки результати не задовольнить його.



Рис. 3.7 - Результат роботи в режимі аналізу для незадовільного прогнозу

3.2 Інтерфейс системи підтримки прийняття рішень

На основі розробленого математичного та алгоритмічного забезпечення для зручного перерахунку даних і практичної перевірки запропонованої методики планування при розробці ПЗ була розроблена система, що дозволяє керівникам ін оектних команд здійснювати оцінку ймовірності виконання проектів до заданого терміну.

Крім безпосередньо практичної реалізації моделі системи масового обслуговування, відповідно до типовою схемою ведення проекту, представленої у другому розділі, реал ізовал також наступні функції:

Модуль формування команди (розділ «Команда», рис. 3.8). Модуль призначений для завдання інформації про команду розробки, автоматичного коригування відомостей про продуктивність команди, в залежності від виконаних завдань, на основі якої перераховується ймовірність виконання інших завдань. А також для ручного коректування складу і продуктивності розробників.

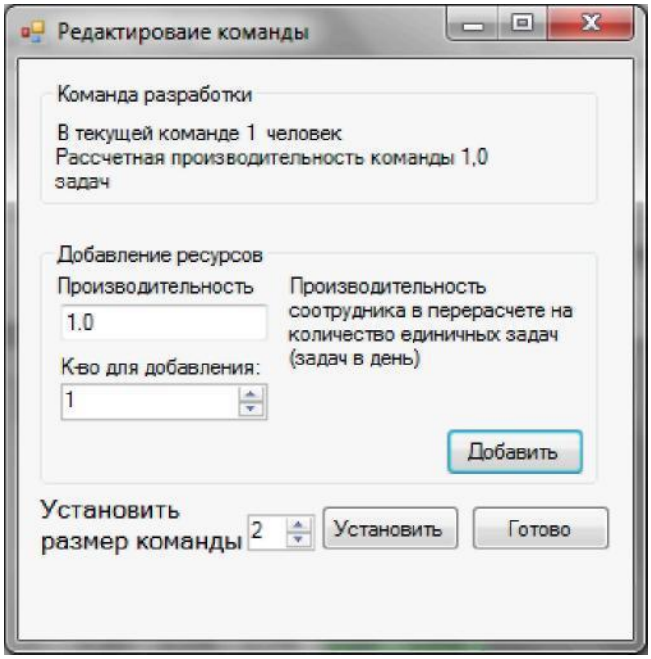


Рис. 3.8 - Розділ управління командою

Модуль формування списку завдань (рис. 3.9). Як було сказано у другому розділі роботи, завдання проекту є граф. Модуль дозволяє побудувати граф завдань, а потім зробити його «обхід», сформувавши список задач з урахуванням пріоритетів.

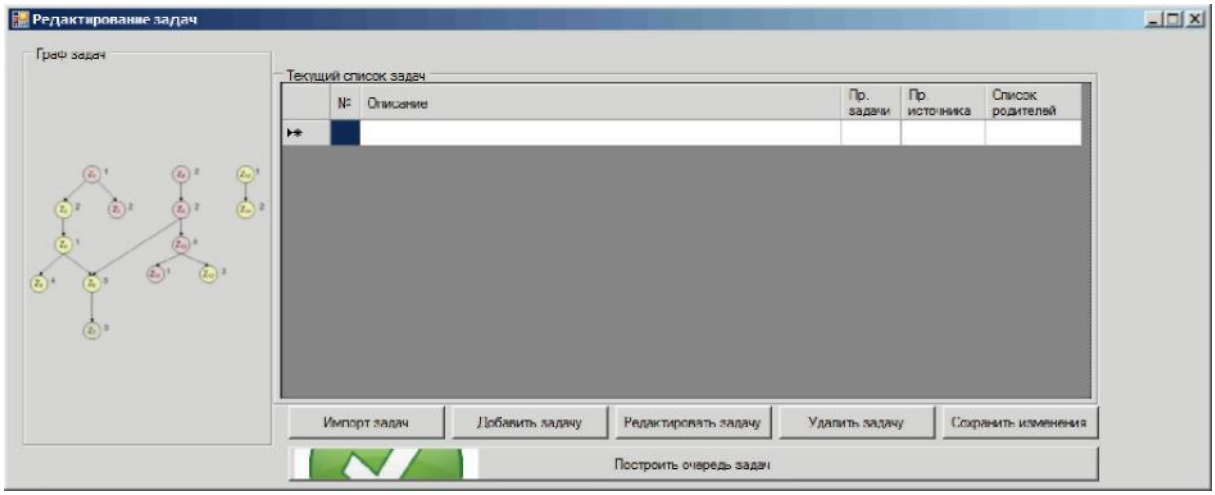


Рис. 3.9 – Вікно формування списку завдань

Модуль прогнозування. (рисунок 3.10) Основний модуль програми. У модулі реалізована пропонована математична модель, з допомогою якої програма р ассчітивает ймовірність виконання заданого кількості завдань за ітерацію, а також пропонує керівникові проекту можливі варіанти підвищення цієї ймовірності, в разі якщо вона нижче порогової.

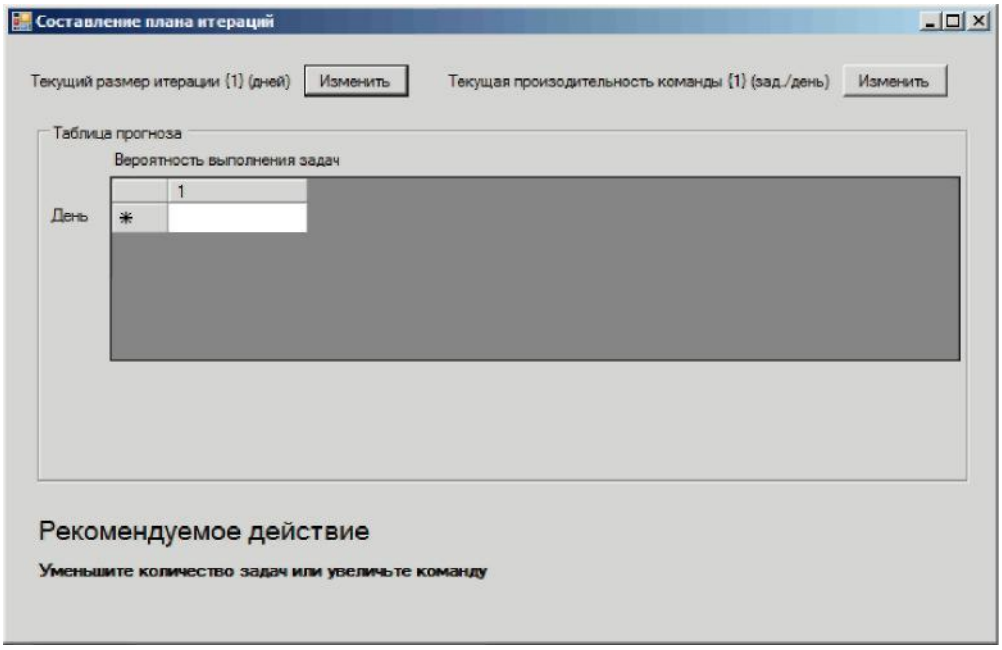


Рис. 3.10 - Модуль прогнозування

В самому успішному випадку, після внесення завдань за проектом, користувачеві необхідно тільки фіксувати виконання завдань, така функція доступна в меню головного вікна системи (рісунок3.11).

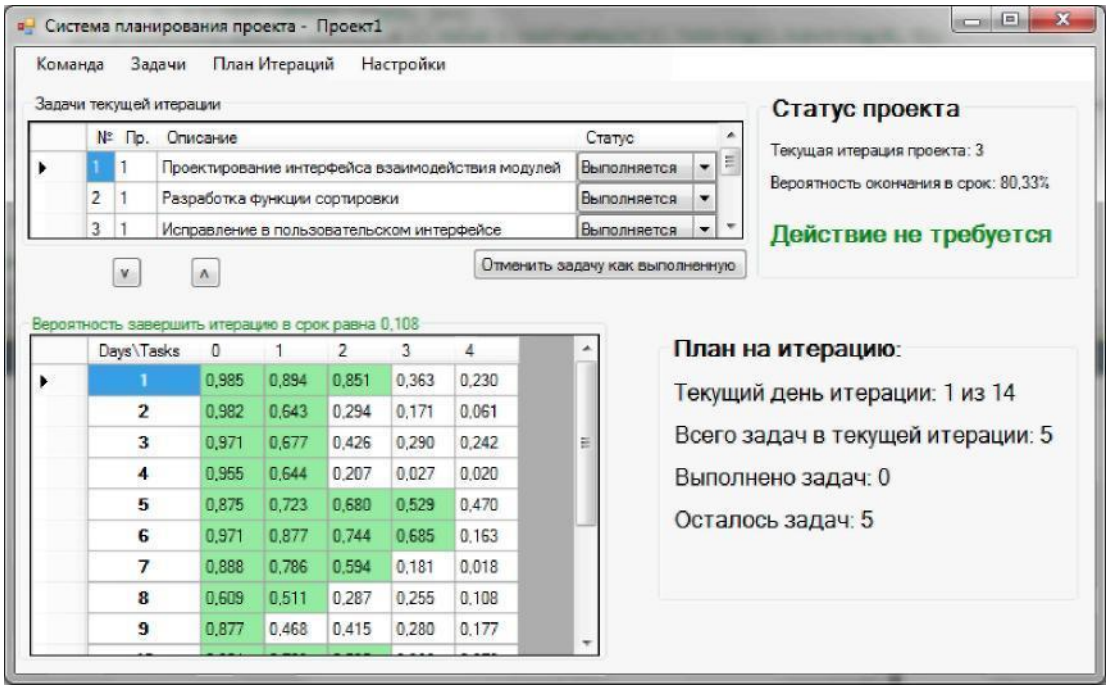


Рис. 3.11 - Головне вікно

3.3 Прогнозування процесу виконання завдань в проекті розробки системи автоматизованого аналізу текстів

3.1.3 Опис проекту

В ході експерименту було виконано планування і реалізація завдань ітерації розробки в проекті створення системи автоматизованого аналізу тексту,

розробка якої ведеться на кафедрі «Проектування обчислювальних комплексів» НДУ МАІ. Вибір обумовлений тим, що

• розміри проекту дозволяють мати достатній ступінь свободи по розподіленню завдань на ітерації і відбору завдань;

• проект ведеться вже кілька років, в ньому накопичилося безліч завдань, пов'язаних з розробкою функціоналу, за яким вже практично не потрібно проводити додаткові дослідження. дослідницькі роботи мають істотну творчу складову, і не можуть бути достовірно оцінені;

• команда розробки проекту складається з фахівців різної кваліфікації , що дозволяє простежити залежності ймовірності виконання завдань при різних конфігураціях команди розробки;

• важлива також відкритість розробки даного проекту, на відміну від більшості комерційних проектів, розкриття деталей яких неможливо через юридичних обмежень.

Система аналізу тексту призначена для обробки текстової інформації, використовує останні напрацювання в області програмних засобів, алгоритмів і методів автоматизованого аналізу тексту. Система має постійно поповнюємо ий набір гнучких інструментів і можливостей впливу на алгоритми обробки текстів.

Система аналізу розвивається вже е кілька років, за цей час значно збільшився її розмір, як наслідок збільшилася кількість завдань, необхідних для реалізації, треба було залучення більшої кількості розробників. Також виникла потреба в застосуванні будь-якого з відомих підходів до розробки ПЗ. Гнучкий підхід дуже добре підходить для інформаційних систем, в яких вже є базова частина, що включає ядро системи і основні інтерфейси взаємодії між модулями, оскільки в цьому випадку вдається подолати, мабуть, ос новной недолік «гнучких» підходів, а саме відсутність добре продуманої і документованої архітектури розроблюваних систем. На кожну ітерацію виноситься той чи інший набір функцій, відповідно до потреб користувача, а сі стема повністю залишається «За рамками», і після якоїсь певної точки стає занадто великовагової, повної різних і навіть неузгоджених частин, нездатною до подальшого розвитку. Однак, коли архітектура вже опрацьована, гнучкі підходи покази вают велику ефективність за рахунок більш тісної взаємодії всередині команди і з замовником. Тому для подальшого розвитку системи аналізу тексту було прийнято рішення застосовувати гнучкий підхід до розробки, а розраховувати і відбирати завдання до аждой ітерації за допомогою пропонованої моделі.

3.1.3 Формування загального списку завдань за проектом

Для побудови завдань для системи проводиться збір вимог і побажань замовника IT-проекту. На наступному кроці встановлюються залежності завдань і пріоритети кожного конкретного завдання.

Після цього етапу виходить не оцінений чорновий варіант графа завдань проекту. По ньому можна провести коригування пріоритетів завдань, з урахуванням того факту, що ніяке підпорядкована завдання не може бути пріоритетнішим за батьківське завдання. Таке коригування виконується по кнопці «Скорегувати». У програмі реалізований рекурсивний алгоритм обходу графа завдань проекту, список завдань представляє собою двусвязний список, у кожного елемента в загальному випадку може бути *N* батьків і *K* залежних елементів. Також є можливість автоматичного проставлення пріоритету батьківської або підпорядкованої завданню, рівного пріоритету пов'язаної з нею завданням, в сл учае якщо завдання пріоритету не має.

Найскладніший і викликає розбіжності крок - оцінка вимог. Пропонована модель працює з уже готовими оціненими вимогами, і велика перевага моделі в тому, що вона не вимагає точних часових оцінок для великих завдань, розрахункові величини коригуються по ходу виконання проекту. У моделі істотною є необхідність коректної взаємної оцінки різних вимог, тобто якщо завдання більше і складніше будь-якої з наявних завдань, то вона мала б ть оцінена в потрібне число разів більшим значенням. Незважаючи на те, що в системі вказані трудодні, насправді можна застосовувати будь-які відомі одиниці часу. Наприклад, за одиницю часу може бути прийнято час виконання еталонної задачі. Згодом (протягом 1-2 ітерацій) в системі продуктивність праці буде вирівняна під цей стандарт. такий спосіб оцінки трудовитрат позволяє т піти від складного питання «За скільки буде виконана задача?» до більш простому «У скільки разів задача Б складніше завдання А?» .

Провівши оцінку завдань, можна отримати список завдань з розставленими пріоритетами.

Цей список вже може бути пригод ен до початку виконання, при необмежених ресурсах і вартості розробки, однак для отримання оцінок потрібно скласти план завдань і виділити ітерації, що задовольняють вибраним критеріям успішності.

Для побудови плану ітерацій, проведення розрахунків в ероятностей виконання завдань потрібно з цього списку завдань з пріоритетами скласти чергу одиничних завдань. Будь-яке завдання може бути декомпозирована на більш дрібні завдання, аж до сукупності декількох одиничних завдань, наприклад, що реалізують окремими е методи. У програмі одиничної будемо вважати завдання, яку один фахівець може виконати за один робочий день.

Програма виконує цю операцію автоматично, після завершення попереднього етапу при переході на форму «планування ітерації». Важливим мом ентом є збереження зв'язків одиничних завдань між собою, програма враховує, що не можна в одну ітерацію запланувати не ціла кількість завдань з початкового списку. Форма планування ітерації представлена на малюнку 3.12.

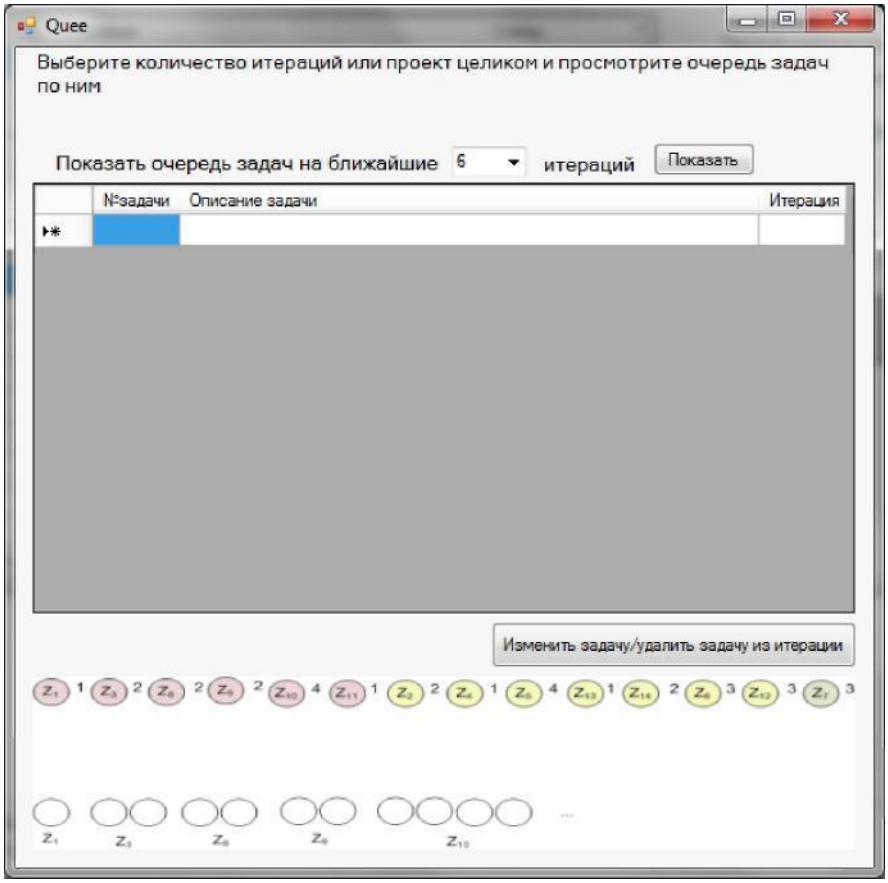


Рис. 3.12 - Черга завдань

3.3.3 Формування плану проекту

З Адачі складання плану ітерацій вирішується за допомогою форми складання плану ітерацій, представленої на малюнку 3.13.

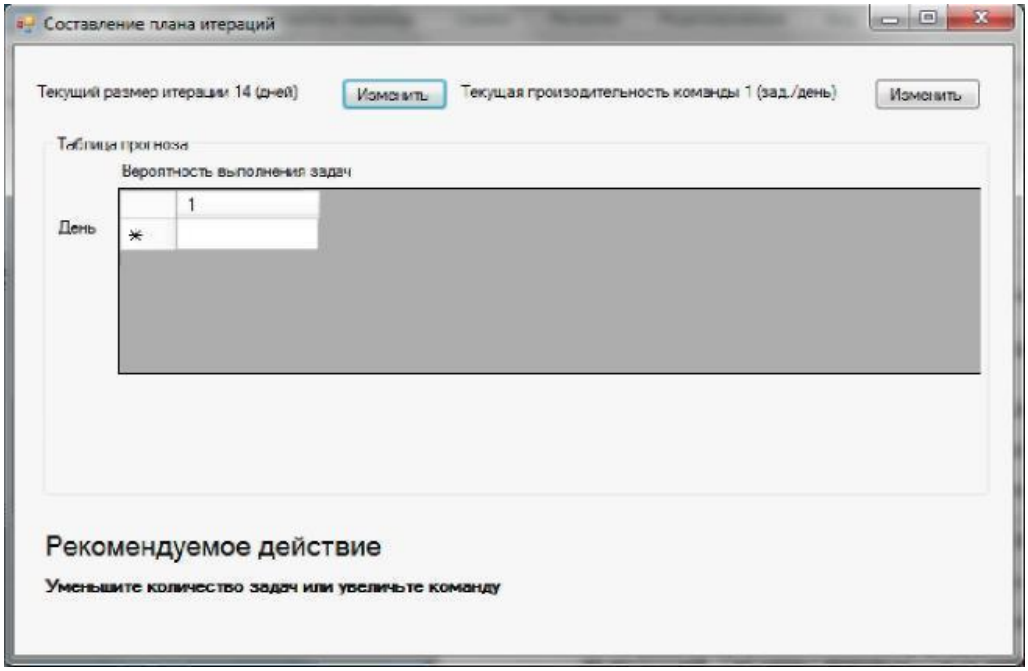


Рис. 3.13 - Складання плану ітерацій

Для кожної ітерації автоматично визначається число виконуваних задач і за допомогою запропонованої моделі обчислюється ймовірність виконання їх до закінчення ітерації. Для керівника проекту є можливість в разі потреби вносити корективи в план ітерації, змінюючи одне або відразу два обмеження: за часом або за кількістю завдань для кожної Ітера ації. Така гнучкість необхідна, оскільки на практиці існують важливі завдання проекту, наприклад, реалізація пріоритетних бізнес сценаріїв, в иполненіе яких не повинно бути зірвано.

3.3.4 Контроль ходу виконання я проекту

Кожен проект в більшій чи меншій мірі відхиляється від початкового плану, тому створена система дозволяє відслідковувати такі зміни (наприклад, завдання виконано раніше або пізніше терміну), що дозволяє керівнику проекту оперативно реагувати на такі відхилення. Загальний вигляд форми контролю ходу проекту представлений на малюнку 3.13.

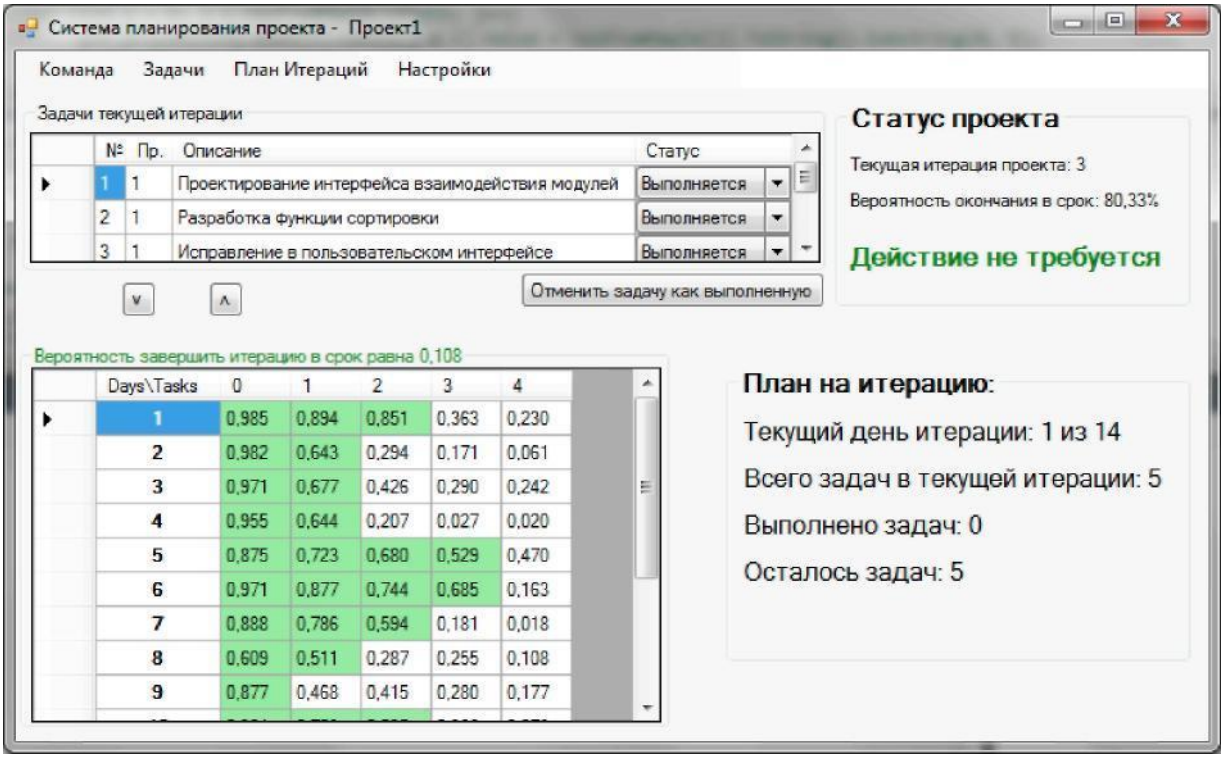


Рис. Форма статусу проекту

Автоматично кожен день або по завершенню завдання пересчіти ється ймовірність виконання поточної ітерації розробки проекту в намічений термін. Важливо розглянути в першу чергу ситуації, коли завдання з будь- яких причин не виконана в запланований термін. При невиконанні однієї або декількох задач на початок наступного дня загальна результуюча ймовірність завершення ітерації може не зменшитися нижче заданого порогу, в цьому випадку не потрібно зміни поточного плану ітерації. Якщо ж ймовірність стала нижче даного порогу, то це є сигналом для прий ку заходів. У першому розділі були проаналізовані різні підходи до ведення проекту, в цьому випадку вони також справедливі:

• заходи, засновані на обмеженні ресурсів. Зазвичай в таких випадках мова йде в першу чергу про час. До таких заходів можна віднести у меньшеніе обсягу робіт. Зменшення обсягу робіт, вказане керівником проекту за довго до його завершення, як правило, не тягне за собою конфліктних ситуацій і сприяє більш тісному співробітництву з замовником. Це в кінцевому підсумку вигідно як замовнику , тому що для нього проект є більш прозорим і передбачуваним, так і розробникам, які мають можливість більш ретельно опрацювати вимоги.

• заходи, засновані на обмеженні обсягу робіт. Незважаючи на те, що гнучкі підходи до розробки ПЗ тр ебують наявності незмінною тривалості ітерацій, на практиці, для виконання завдань довжини ітерацій можуть бути не сильно змінені (зазвичай в більшу сторону) без значної шкоди для всього проекту.

• організаційно-адміністративні заходи. Їх необхідно ви делять в окрему групу, оскільки завдання виконуються людьми, а значить можна не нехтувати впливом людського фактора на розробку ПЗ . Різні аспекти теорії менеджменту, такі як способи підвищення продуктивності праці, підвищення мотивації , відповідальності співробітників, а також збільшення ефективності проектування і програмування виходять далеко за рамки даної роботи. Однак в ході натурного експерименту вдалося лише за рахунок таких заходів підвищити продуктивність роботи окремих членів команди в 2-2.5 рази, тільки за рахунок додаткового навчання після стажування такого різкого зростання проізводітельнос ти досягти практично неможливо.

4 .3.5 Аналіз результатів прогнозу і вибір параметрів проекту Форма поточного статусу проекту в поєднанні з іншими формами може бути використання для моделювання впливу різних нестандартних ситу ацій на проект. Необхідно перевірити ситуації, пов'язані з впливом на календарний план проекту розширення наявних вимог замовника. Залежно від своїх масштабів таке розширення може призвести як до незначного зменшення ймовірності виконан ня проекту в термін, так і до суттєвого порушення поточного календарного плану. Система для швидкого моделювання ступеня впливу на проект і ймовірності його закінчення в намічений термін дозволяє з достатньою точністю перевірити, наскільки ті чи інші і Зміна в обсягах робіт можливі без перегляду всього проекту. З його допомогою також можна оперативно провести попереднє планування оновленого проекту для подальшого вирішення про доцільність внесення змін.

Друга часто зустрічає потрібно було сть перегляду термінів і плану проекту з'являється в зв'язку зі збільшенням розміру завдань щодо початкового плану. Практично будь-яке завдання розробки навіть невеликий функції, тим більше, якщо початкова оцінка була зроблена для більших вузлів сис теми, може містити додаткові складності, які неможливо врахувати на початкових етапах. Система реагує на подібні ситуації з двох сторін. По-перше, наявність невиконаних завдань за минулий період з невеликими застереженнями може бути розглянуто як поява нових і впливає на зміну ймовірності планового завершення поточної ітерації і всього проекту. По-друге, невиконані завдання свідчать про зменшенні продуктивності праці кому анди (інтенсивності виконання

задач), на основі якої здійснюється розрахунок ймовірностей в запропонованій моделі.

Поряд з неповною оцінкою завдань також необхідно розглядати ситуації, пов'язані з втратою розробника в команді і, навпаки, вплив посилення команди на проект. Зменшення кількості працюючих над проектом людей, крім причин

хвороби або звільнення співробітників, може бути викликано перекладом (або частковим перекладом) на інші проекти. З іншого боку, збільшення числа людей з-за підвищена шення важливості проекту за рахунок залучення людей з інших проектів також не рідкість. Це характерно в першу чергу для великих компаній, де одночасно ведеться розробка декількох різних проектів, і в залежності від потреб, термінів, пріоритетів в кожного проекту наявні ресурси можуть бути
перерозподілені. Розроблена програма за допомогою форми «Команда розробки» дозволяє оперативно зробити прорахувати вплив на календарний план проекту збільшення і зменшення кількості членів команди разр аботкі.

Будь системі ведення проектів потрібні максимально точні оцінки завдань, наприклад, при використанні методу PERT , в залежності від набору оцінених завдань, може бути різна форма критичного шляху, а значить, все оцінки проекту можуть змінитися через одну або кількох неточних оцінок. У розробленій системі цей недолік не виявляється, неточні оцінки конкретних завдань впливають тільки на виконання цих завдань, а систематичні неточні оцінки призводять до зміни інтенсивності рішення зад ач командою, яка автоматично перераховується в залежності від поточних виконаних завдань, а прогноз на наступну ітерацію будується з використанням оновленої продуктивності.

Як було сказано вище, в ході виконання проекту, програма автоматичні ки щодня перераховує ймовірності рішення різного числа завдань до кінця ітерації. На малюнку 3.15 показано стан на перший день чергової ітерації.

З наведених на малюнку 3.15 даних видно, що вероятнос ть виконання в строк всіх запланованих завдань менше заданого порогового значення (0.8765 <0.9). Тому проводиться розрахунок максимальної кількості завдань, які можуть бути виконані за відведений на ітерацію час з достатньою ймовірністю.

На практ ике часто необхідно вирішити всі поставлені завдання, тому актуальним завданням є отримання прогнозу, за який час можуть бути виконані всі поставлені завдання з задовільною ймовірністю.

Таким чином, для даної команди розробки виконання всіх поставлених завдань на ітерацію можливо з імовірністю 87.7%, з достатньою ймовірністю 91,8% можливе виконання трьох завдань за відведений час ітерації (3 дні) або виконання всіх поставлених завдань за 5 днів з імовірністю 95.5% .

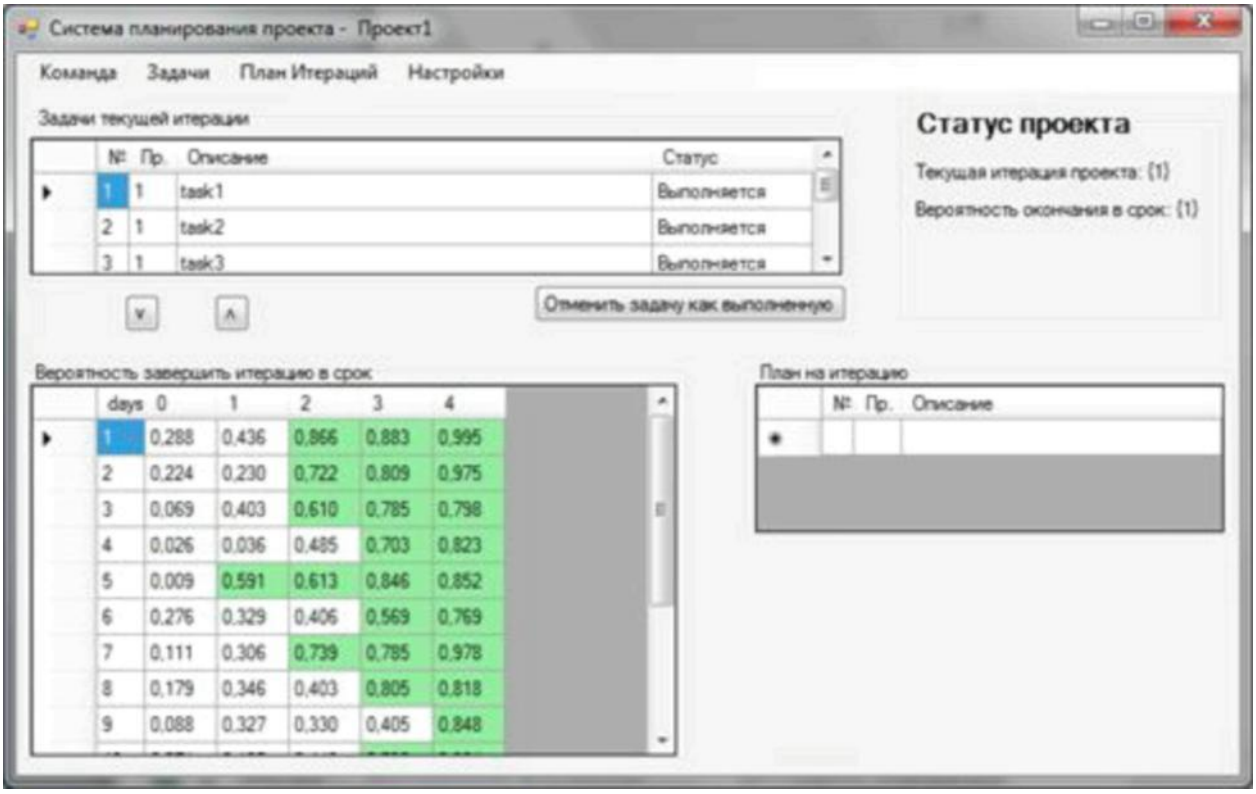


Рис. 3.15 - Стан проекту на перший день ітерації

3.4 Подальше використання розробленої системи підтримки прийняття рішень

Програма передбачена для моделювання розробки одного проекту, і інші можливі проекти організації враховуються тільки на етапі завдання ресурсів, зайнятість яких може бути частковою. Додавання підтримки портфелів проектів або програ мм проектів дозволить застосовувати розроблену систему і там, де необхідно враховувати зв'язку між проектами.

Разом з додаванням підтримки безлічі проектів для прогнозування також зростає роль бюджетного планування. Таким чином, з'являться можливості моделювання ходу ведення проекту, наприклад, з урахуванням мінімізації витрат. Це може бути зроблено без істотної перебудови поточної системи за рахунок додавання кожному розробнику умовної вартості його трудодня і разом з розрахунком ймовірності ви конання завдань обчислювати їх «вартість». Очевидно, що при зменшенні вартості дня розробки на 50% при збільшенні термінів на 25-30% загальна вартість проекту зменшиться.

Другим напрямком використання і розвитку системи є її інтеграція з суті ющими системами ведення проекту, обліку трудовитрат, завдань, бюджету, і т.д. Розуміючи, що конкуренцію, наприклад, з MS Project або аналогами вінтерфейсній складової, а також в частині використання спільно з іншими системами підприємства, виграти небольш ними силами неможливо, найбільш

ефективним шляхом може бути інтеграція з існуючими промисловими системами ведення проектів. Для багатьох з них існує можливість надати окремий модуль прогнозування, в тому чис ле використовуючи власний інтерфейс взаємодії з користувачем або задіявши наявний в разі його успішної адаптації. такий спосіб дає великі можливості для застосування розробленої програми в існуючих прое ктах, оскільки не потрібно перехід на нові програмні інструменти. Користувачі в цілому більш охоче приймають ті програми, які їм уже знайомі.

Згодом, коли в організації буде накопичена база виконаних проектів, в розробленій системі буде потрібна можливість зручного графічного представлення накопиченої статистичної інформації, наприклад, про збіг прогнозів і результатів для більш точного визначення порогової ймовірності прийняття завдання в ітерацій ю і побудови графічних звітів про результати виконання проектів для аналітиків.

Розроблена система може бути використаний для створення системи з веб-інтерфейсом для ведення проектів. Крім очевидних переваг в мобільності і та спользование тонких клієнтів, це дозволить простіше вести централізовану базу проектів і прогнозів. Поточна архітектура системи передбачає легкість використання її ядра для створення інструменту з веб- інтерфейсом, тому що в розробленій системі воно відділений про від інтерфейсної частини, яка може бути змінена як на вбудований модуль, так і на веб- інтерфейс.

3.5 Висновки

1. Розроблено архітектуру системи підтримки прийняття рішень при управлінні проектами створення ПЗ і оцінки термінів успішного завершення я проектів.

2. Розроблено архітектуру, описані вхідні і вихідні дані і принципи функціонування модуля побудови і аналізу графа завдань і створення на його основі черзі завдань.

3. Розроблено систему аналізу і прийняття рішень при управлінні проектами розробити конструкцію програмного забезпечення.

4. Наведені приклади роботи створеної системи, що використовує запропоновану модель при плануванні і виконанні проекту розробки системи автоматизованого аналізу текстів.

4. Проаналізовано методи коригування плану прое кту і їх відображення в програмі в разі відхилення проекту від наміченого плану або при появі раніше неврахованих нових завдань.

6. Запропоновано основні напрямки використання системи підтримки прийняття рішень як модуля більшої системи або в якості самостійної системи з web -інтерфейсом.

ВИСНОВОК

На основі сучасних методів системного аналізу, отримані результати для підвищення точності оцінки часу, необхідного для виконання ІТ-проектів.

1. Проведено аналіз сучасних підходів до ведення ІТ-проектів , основних методів прогнозування ходу виконання проектів та їх програмні реалізації, виділені основні етапи виконання ІТ-проекту: ініціація проекту і складання списку завдань, побудова плану ітерацій проекту, виконання завдання по проекту.

2. Запропонована функціонально-інформаційна модель процесу виконання ІТ-проекту з розбивкою етапів проекту на фази з точки зору складу і структури оброблюваної інформації і виконуваних над нею операцій, що дозволяє провести аналіз процесів перетворення даних в ході управління ІТ-проектом і виявити можливості автоматизації.

3. Проведено аналіз процесів перетворення даних при виконанні ІТ-проекту і запропоновані способи автоматизації процесу перетворення даних на етапах планування та виконання завдань в доповнення до існуючих засобів автоматизації управління вимогами і завданнями. За результатами аналізу запропоновано підхід до ведення ІТ-проекту з використанням системи підтримки прийняття рішень, що дозволяє до початку і в процесі виконання проекту прогнозувати хід його виконання.

3. Розроблено модель процесу виконання завдань командою ІТ-проекту, що дозволяє в залежності від кількості членів команди автоматично адаптувати модель і перераховувати залежності ймовірностей виконання заданої кількості завдань.

5. Розроблено ПЗ системи підтримки прийняття рішень при плануванні ІТ-проектів, що забезпечує визначення характеристик проекту і оптимальне поєднання часу, ресурсів і кількості задач з високою точністю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Зачко О. Б., Івануса А.І., Кобилкін Д.С. Управління проектами: теорія, практика, інформаційні технології. – Львів: ЛДУ БЖД, 2019. – 173 с.
2. Катренко А.В. Управління ІТ-проєктами. Книга 1. Стандарти, моделі та методи управління проєктами: підручник. Львів: «Новий Світ- 2000». 2017. 550 с.
3. Ноздріна Л., Ящук В., Полотай О. Управління проєктами. Київ Центр навчальної літератури, 2020. 432 с.
4. Хігні Д. Основи управління проєктами. Харків: Фабула, 2020. 272 с.
5. Микитюк П. П., Брич В. Я., Микитюк Ю. І., Труш І. М. Управління проєктами: Підручник. Тернопіль, 2021. 416 с.
6. Білас О.Є. Якість программного забезпечення та тестування. Навчальний посібник /О.Є. Білас – Львів: Вид-во Львівської політехніки, 2021. 216 с.
7. Якість програмного забезпечення та тестування: базовий курс. Навчальний посібник /За ред. Крепич С.Я., Співак І.Я. /для бакалаврів галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення». Тернопіль: ФОП Паляниця В.А., 2020. 478 с.
8. Гвоздь М.Я., Злидник Ю.О. AGILE – нова методологія менеджменту: теоретичні аспекти. Електронний журнал «Інфраструктура ринку». 2018. No25. С. 230-234.
9. Основи управління ІТ проєктами: навч. посіб. для студ.спеціальності 122«Комп’ютерні науки» /КПІ ім. Ігоря Сікорського ; уклад.: В. О.Кузьміних, Р. А. Тараненко. Електронні текстові дані (1 файл:1,998 Мбайт). Київ: КПІ ім. Ігоря Сікорського, 2019. 75 с.
10. Управління ІТ-проєктамив Microsoft Project: Комп’ютерний практикум: навчальний посібник для студентів спеціальності 122 “Комп’ютерні науки” для всіх спеціалізацій /Л. М. Добровська, О. В.Аверьянова; КПІ ім. Ігоря Сікорського. Електронні текстові данні (1 файл: 17,6 Мбайт). Київ: КПІ ім. Ігоря Сікорського, 2020. 152 с. URL: https://ela.kpi.ua/bitstream/123456789/33622/1/KV_prakt_Upr_proektamy.pdf
11. Фурсова Н.А., Кириченко Ю.В. Аналіз сучасних програмних інструментів управління ІТ - проєктами/ Тези 69-ої наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів університету. Том 1. (Полтава, 19 квітня –19 травня 2017 р.) Полтава: ПолтНТУ, 2017, с. 221-223.
12. Грицюк Ю.І., Андрущакевич О.Т. Засіб для визначення якості програмного забезпечення методами метричного аналізу /Науковий вісник НЛТУ України, 2018, т.28, No6. с.159 – 170.

13. 12 кращих методологій розробки програмного забезпечення з перевагами та недоліками. URL: https://www.smart-it.com/uk/2021/08/12-best-software-development-methodologies-with-pros-and-cons/
14. Agile Manifesto. URL: http://agilemanifesto.org/ 3. Відео-курс Scrum Master Fundamentals - навчання основ Scrum і Agile гнучких методологій. URL: https://www.youtube.com/playlist?list=PLhZAXRyXTJpOUkbI2H2M9n__A9PxzQEi-4.
15. Гнучка методологія. Найкраща практика у 2023 році. URL: https://ahaslides.com/uk/blog/agile-methodology/
16. Давід Дж. Андерсон. Канбан. Успішні еволюційні зміни для вашого технологічного бізнесу. Видавництво Фабула. 2021. https://books.google.com.ua/books?id=xFY9EAAAQBAJ&hl=uk&source=gbs_navlinks_s
17. Екстремальне програмування та його роль у розробленні ПЗ. URL: https://foxminded.ua/ekstremalne-prohramuvannia/
18. Екстремальне програмування (XP). URL: https://worksection.com/ua/blog/extreme-programming.html
19. Інсталяція Redmine на Ubuntu. URL: https://bunyk.wordpress.com/2010/07/26/redmine-ubuntu-install/
20. Redmine. URL: https://www.redmine.org/
21. Методології розробки програмного забезпечення: поняття, принципи, методи та етапи розробки. URL: http://nauka-tehnika.com.ua/news/9337/
22. Настанова до зводу знань з управління проєктами та Стандарт управління проєктами (Настанова PMBOK, сьоме видання). – URL: https://pmiukraine.org/wp-content/uploads/2022/08/PMBOK7_Ukr_ForPersonalUseOnly.pdf
23. Перит І. О. Перспективи впровадження Kanban в управління бізнесом вітчизняних суб'єктів господарювання. Бізнес Інформ. 2019. No 8. С.218-228. URL: http://nbuv.gov.ua/UJRN/binf_2019_8_31 .
24. Смолич Д.В. Інноваційні методи управління проєктами / Д. В. Смолич // Економічний форум. 2019. No 4. С. 50-54.
25. Управління ІТ-проєктамив Microsoft Project: Комп’ютерний практикум: навчальний посібник для студентів спеціальності 122 “Комп’ютерні науки” для всіх спеціалізацій /Л. М. Добровська, О. В.Аверьянова; КПІ ім. Ігоря Сікорського. Електронні текстові данні (1 файл: 17,6 Мбайт). Київ: КПІ ім. Ігоря Сікорського, 2020. 152 с. URL: https://ela.kpi.ua/bitstream/123456789/33622/1/KV_prakt_Upr_proektamy.pdf
26. ChatGPT. URL: https://chat.openai.com/auth/login
27. Ken Schwaber & Jeff Sutherland. The Scrum Guide, 2020. URL: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf .
28. Що таке Канбан. URL: https://brainrain.com.ua/uk/shcho-take-kanban/

29. Що таке SCRUMBAN: основи. URL: https://iampm.club/ua/blog/shho-take-scrumban-osnovi/
30. Crystal Clear методологія - простота і досконалість для бізнес-додатків. URL: https://worksection.com/ua/blog/crystal-clear.html
31. DSDM методологія. URL: https://wiki.cuspu.edu.ua/index.php/DSDM_%D0%BC0%B3%D1%96%D1%8F .
32. Lean Software Development. URL: https://learn.microsoft.com/en-us/previous-versions/redirectedfrom=MSDN
33. Zosym Maxym. Crystal Agile Framework. URL: https://www.maxzosim.com/crystal-agile-framework/
34. Реінжиніринг бізнес-процесів: зміст та основна характеристика. URL: https://library.if.ua/book/28/1898.html
35. Basecamp. URL: https://basecamp.com/
36. Trello. URL: https://trello.com/uk
37. Worksection. URL: https://worksection.com .
38. Як ставити та досягати цілей за допомогою методики SMART. URL: https://yaktak.com.ua/yak-staviti-cili-za-metodikoyu-smart/
39. Що таке SMART-цілі. URL: https://info.nic.ua/uk/blog-uk/smart-targets-2/
40. Цілі SMART: 10 прикладів + інструкція від ТОП-менеджера. URL: https://www.kozaky.org.ua/cili-smart-10-prikladiv-instrukciya-vid-top-menedzhera/
41. Принципи та види бенчмаркінгу. URL: https://pidru4niki.com/74017/marketing/printsipi_vidi_benchmarkingu
42. Модель зрілості спроможностей (Capability Maturity Model – CMM). URL: https://www.maxzosim.com/modiel-zrilosti-spromozhnostiei/
43. Інтеграція моделі зрілості вимог (CMMI). URL: https://visuresolutions.com/uk/requirements- management-traceability-guide/cmmi/
44. Методології розробки програмного забезпечення: поняття, принципи, методи та етапи розробки. URL: http://nauka-tehnika.com.ua/news/9337/
45. Freeplane . URL: https://docs.freeplane.org/
46. Ambler, S., & Lines, M. Choose your WoW: A Disciplined Agile Delivery Handbook for Optimizing Your Way of Working. Project Management Institute, 2020. 445 p.
47. A Guide to the Project Management Body of Knowledge (PMBOK® Guide). Seventh Edition: Project Management Institute, 2021. 250 p.
48. Beck, K., Fowler, M. (2014). Planning extreme programming. Upper Saddle River, NJ: AddisonWesley. 160 p.
49. David J.Anderson. KANBAN. Successful Evolutionary Change for Your Technology Business. Copyright 2010 by David J.Anderson. 278 p.

50. Dionisio C.S. Microsoft Project 2019 For Dummies John Willey & Sons, Inc., 2019. 352 p.