

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК  
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

## КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Моделі адаптивного управління людино-машинною взаємодією в інформаційних системах»

за спеціальністю 122 «Комп'ютерні науки»,  
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ.м-81 Федорова Анастасія Валеріївна

Кваліфікаційну роботу  
захищено на засіданні ЕК  
з оцінкою \_\_\_\_\_

«\_\_\_» грудня 2019 р.

Науковий керівник \_\_\_\_\_

(підпис)

д.т.н., проф., Лавров Є.А.

Голова комісії \_\_\_\_\_

(підпис)

Шифрін Д.М.

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів  
без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Суми-2019

**Сумський державний університет**  
**Факультет електроніки та інформаційних технологій**  
**Кафедра комп'ютерних наук**  
**Секція інформаційних технологій проектування**  
**Спеціальність 122 «Комп'ютерні науки»**  
**Освітньо-професійна програма «Інформаційні технології проектування»**

**ЗАТВЕРДЖУЮ**

Зав. секцією ІТП

\_\_\_\_\_ В. В. Шендрик  
«\_\_» \_\_\_\_\_ 2019 р.

## **ЗАВДАННЯ**

**на кваліфікаційну роботу магістра студентіві**

*Федорова Анастасія Валеріївна*

(прізвище, ім'я, по батькові)

**1 Тема проекту** *Моделі адаптивного управління людино-машинною взаємодією в інформаційних системах*

затверджена наказом по університету від «19» листопада 2019 р. №2305-III

**2 Термін здачі студентом закінченого проекту** « 10 » грудня 2019 р.

**3 Вхідні дані до проекту** \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)** \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)**

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

### 6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання \_\_\_\_\_.

Керівник \_\_\_\_\_  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

### КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка

Магістрант \_\_\_\_\_

Федорова А.В.

Керівник роботи \_\_\_\_\_

д.т.н., проф. Лавров Є.А.

## РЕФЕРАТ

Тема кваліфікаційної роботи магістра « Моделі адаптивного управління людино-машинною взаємодією в інформаційних системах».

Пояснювальна записка складається зі вступу, 4 розділів, висновків до розділі та загального, списку використаних джерел із 34 найменувань, 5 додатків.

Загальний обсяг роботи – 109 сторінок, у тому числі 63 сторінки основного тексту, 3 сторінки списку використаних джерел, 43 сторінок додатків.

Кваліфікаційну роботу магістра присвячено розробці моделей для реалізації механізмів адаптивного управління людино-машинних систем.

В роботі проведено дослідження, яке дозволило описати матриці ймовірностей для станів та переходів, які потім були використані для оцінки оптимальних варіантів інтерфейсу для різних груп користувачів

У роботі виконано розподіл цільової аудиторії за допомогою факторного аналізу, який також дозволяє виявляти найслабші місця системи відповідно до 4х факторів.

Результатом проведеної роботи є побудована математична модель, яка допоможе ефективно здійснювати пошук оптимальних вершин переходів. Практичне значення роботи полягає у

- розробці узагальненої моделі інформаційної системи
- проведенні оцінки рівня задоволеності користувачів інформаційною системою
- створенні моделі для вибору варіантів людино-машинної взаємодії для реалізації адаптивного управління
- проведенні апробації моделей

Ключові слова: оптимізація, математична модель, ергономіка, задачі Маркова, прототипування , юзабіліті.

## ЗМІСТ

Вступ.....	6
1 Аналіз предметної області.....	7
1.1 Проблеми usability сайту.....	7
1.2 Основні методи юзабіліті-тестування програмних інтерфейсів .....	10
1.3 Аналіз програм-аналогів для створення прототипів веб-сайтів.....	22
Висновки до розділу 1 .....	25
2 Постановка задачі та методи дослідження .....	26
2.1 Мета та задачі .....	26
2.2. Методи дослідження.....	26
2.3 Методи реалізації математичної моделі.....	31
2.4. Планування робіт .....	33
Висновки до розділу 2 .....	33
3. Моделювання системи адаптивного управління.....	34
3.1 Функціональне моделювання.....	34
3.2 Моделювання діаграми IDEF3.....	37
3.3 Розробка графу системи .....	38
Висновки до розділу 3 .....	40
4. Реалізація математичної моделі .....	41
4.1 Вибір цільових груп користувачів.....	41
4.2 Факторний аналіз .....	42
4.3 Розробка підходу до розв’язання задачі адаптивного управління .....	44
4.4 Створення прототипів для груп користувачів.....	46
4.5 Реалізація рішення задачі оптимізації.....	52
4.6 Аналіз обраних прототипів .....	59
Висновок.....	63
Список використаної літератури .....	64

Додаток А.Планування робіт .....	67
Додаток Б. Розв'язання в MATLAB .....	72
Додаток В. Прототипи .....	73
Додаток Г Копії публікацій .....	83
Додаток Д Наукові конференції.....	104

## ВСТУП

**Актуальність.** Наростаючий пріоритет використання мобільних пристроїв, а також тенденція підтримки браузером адаптивних ресурсів, призводять до необхідності в адаптивності ресурсів. Однак практика діючих сайтів не враховує всі наростаючі потреби в адаптивності, тому різко знижується юзабіліті цих ресурсів.

**Об'єкт дослідження.** Взаємодія користувачів та інформаційних систем

**Предмет дослідження.** Моделі адаптивного управління людино-машинною взаємодією в інформаційних системах

**Мета.** Розробити моделі для створення адаптивного веб-інтерфейсу для підвищення юзабіліті.

**Практичне значення.** Використання даних моделей дозволить збільшити відвідуваність сайту і сприятиме підвищенню економічної ефективності через збільшення кількості клієнтів і результативності їх роботи.

**Наукова новизна одержаних результатів.** На відміну від існуючих підходів, було запропоновано комплексну модель, яка забезпечує механізм адаптації.

**Апробація результатів:** Результати доповідались на 4 наукових конференціях:

Науково технічна конференція «Інформатика Математика Автоматика», м. Суми: СумДУ , 23-26 квітня 2019

International Scientific Conference «UNITECH'17» - Gabrovo, 17-18 november 2017 - Bulgaria

Науково технічна конференція «Інформатика Математика Автоматика», м. Суми: СумДУ , 5-9 лютого 2018

Науково-практична конференція «Цифрові технології в освіті, науці, суспільстві» м. Петрозаводськ 27–30 листопада 2017

Науково-практична конференція «Цифрові технології в освіті, науці, суспільстві» м. Петрозаводськ , 4–6 грудня 2018

**Публікації.** За матеріалами досліджень опубліковано 6 наукових робіт (перелік та копії додаються). Додаток Г

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Проблеми usability сайту

На сьогоднішній день найбільш актуальною дисципліною в практиці проектування систем і засобів призначеного для користувача інтерфейсу для широкого кола інформаційних систем масового призначення є ергономіка. Особливо популярний напрям розвитку цієї дисципліни, що сформувався під назвою «юзабіліті». Це поняття об'єднує класичні методи інженерної психології з маркетинговою ідеологією і спрямоване на оцінку зручності інтерфейсів для користувачів. Актуальність даного напрямку пов'язана з інтенсивним розвитком техногенного середовища людства, появою Інтернету і необхідністю створення комп'ютерних інтерфейсів для масового користувача.

Особливістю користувачів масових інформаційних систем є відсутність у них систематичної освіти в сфері комп'ютерного програмування і технологій та наявність специфічного досвіду, не пов'язаного з практикою роботи з існуючими комп'ютерними інтерфейсами. Наслідком цього є відсутність або слабкий розвиток у масового користувача навичок роботи з інтерфейсами інформаційних і технічних систем. [10] При цьому число завдань, орієнтованих на широкого користувача, стрімко зростає в зв'язку з загальною комп'ютеризацією і тенденцією заміни людського спілкування в деяких професійних областях на спілкування з комп'ютерними системами. На сьогоднішній день багато сервісів дають можливість аналізу даних будь-якого сайту. Нижче наведено дані для компанії ІТ курсів

Таблиця 1.1. Аналіз відвідувань сайту компанії ІТ курсів

	За день	За місяць
Перегляди	220	6420
Відвідувачі	60	1610
Замовлення	0,37	10



З таблиці 1.1 ми можемо побачити, що при досить успішних числових значеннях кількості відвідувачів, ми маємо доволі низький рівень замовлень через сайт. Такі проблеми можуть виникати, коли користувацький інтерфейс не враховує потреб, занадто ускладнений та містить зайві дії для користувача.

У розряд масових користувачів потрапляють користувачі електронних терміналів з продажу інтернет магазинів та сайтів з замовленнями, оплати комунальних послуг, податків і т.д.

Переваги для компанії, яка звертає увагу на юзабіліті під час процесу забезпечення життєвого циклу включають:

- збільшення продуктивності;
- збільшення продажів і доходів;
- зменшення часу і витрат на навчання;
- зменшення часу і витрат на розвиток;
- зменшення витрат на обслуговування;
- збільшення задоволеності клієнта.

З кожним роком увага до юзабіліті інформаційних продуктів зростає. Все більша кількість розробників програмних засобів намагаються зробити роботу зі своїми продуктами максимально доступною, комфортною та ефективною. Зростає розуміння того, що юзабіліті може стати серйозною конкурентною перевагою в боротьбі за споживача. Технології та типові методи інженерної психології і юзабіліті, що застосовуються у відповідних ситуаціях проектування, повинні бути відомі практичним працівникам. [3] Все це визначає необхідність підготовки фахівців, які володіють методами юзабіліті програмних інтерфейсів інформаційних систем. [1,2].

Фахівці, що займаються розробкою програмного забезпечення, здебільшого не поділяють функціональність використовуваних інструментів з інтерфейсом користувача. Однак призначений для користувача інтерфейс входить в якості елемента в функціональну систему програмного комплексу. З іншого боку, користувачі сайту, програми також можуть не співвідносити реальну функціональність будь-якої програми або комплексу програм з видимим

інтерфейсом. Зручність інтерфейсу для типового користувача стає визначальним фактором в роботі з тим чи іншим сайтом або програмою. Продуманий інтерфейс користувача включає багато аспектів дизайну, налаштовує призначене для користувача сприйняття і ставлення до продукту, яке будується на основі ефективності роботи з інтерфейсами програми або сайту. Інтерфейс користувача містить наступні складові [12]:

1. Запропоновані сайтом або програмою завдання, які відвідувач ресурсу, може виконати в процесі роботи з ним;
2. Основні параметри і складові управління системою;
3. Навігація по сайту або програмі;
4. Дизайн інтерфейсу користувача.

Неякісно розроблений інтерфейс сприяє зниженню функціональності роботи системи і може призвести до відмови або неможливості роботи з сайтом.

Завдання аналізу юзабіліті - виявити проблемні та вузькі місця сайту, з метою спростити навігацію користувача по сайту, максимально повно представити необхідну інформацію і товари, скоротити ланцюжок переходів користувача від стартової сторінки, до сторінки цільової дії, замовлення товару або оформлення заявки. [6]

Якісно проведений аудит юзабіліті дозволяє збільшити конверсію сайту від декількох десятків відсотків, до декількох разів, в залежності від вихідного стану сайту. Процес переходу з відвідувача в покупця візуально представлений у вигляді воронки продажів. В ході аналізу виявляються цілі і потреби користувача, що крім коригування самого сайту може спричинити і коригування маркетингової компанії. Юзабіліті багато в чому базується на людській психології і людських звичках. [5] Знаючи стандарти в області побудови сайтів в необхідному сегменті інтернету, можна швидко виявити помилки в зручності використання ресурсу. Напрямки помилок по web-юзабіліті можна розділити на 3 великі блоки: перший стосується технічних помилок, другий - дизайну, третій - контенту

Визначення або вимірювання якості роботи інтерфейсу вимагає визначення основних параметрів його роботи. Так, на якість роботи сайту можуть впливати такі параметри-метрики (юзабіліті-метрики), як задоволеність, економічність в використанні, ефективність. До параметрів-метрик також слід віднести складові контексту використання і їх підкомпоненти, які піддаються в процесі перевірки оцінюванню.

Аналіз літератури і матеріалів останніх конференцій, присвячених питанням ергономіки, а також практичного стану питання, дозволяє зробити наступні висновки . [4-8]:

1) в існуючих методах підвищення юзабіліті є певні недоліки, пов'язані з їх суб'єктивністю, відсутністю формалізації, способів кількісної оцінки якості інтерфейсу і способів виявлення шляхів поліпшення якості інтерфейсу;

2) в наявному на ринку програмному забезпеченні, що реалізовує ряд розглянутих методик, присутній ряд недоліків, що не дозволяють ефективно виконувати необхідні завдання щодо поліпшення якості інтерфейсу. Незважаючи на наявні результати по розробці методів для забезпечення зручності інтерфейсів веб-додатків і підвищення якості, до теперішнього часу всі ці методи є суб'єктивними і не спираються на формалізовані засоби оцінки. Вони переважно будуються на суб'єктивній думці користувачів та експертів з ергономіки інтерфейсів, а не на математично обґрунтованих методах. Це пояснюється досить високою складністю формалізації критеріїв оцінки, так як серед них присутній психологічний і художній аспект.

## **1.2 Основні методи юзабіліті-тестування програмних інтерфейсів**

В ході проектування бажано оцінити, наскільки якісними є ті рішення, які ви вирішуєте використовувати. Для цього потрібно вийти за межі персонажів і сценаріїв перевірки і запропонувати рішення реальним користувачам. Робити це потрібно тоді, коли рішення набуло певного рівня деталізації, щоб користувачі могли працювати з

ним як з системою, але при цьому є ще достатньо часу, щоб занести виправлення, виходячи з результатів тестування.

Юзабіліті-тести і сеанси, спрямовані на отримання відгуків користувачів, добре підходять для виявлення великих проблем з інфраструктурою взаємодії і для поліпшення таких речей, як написи на кнопках, порядок і пріоритет дій. Вони також життєво важливі для налаштування таких аспектів поведінки, як швидкість прокрутки екрану в якості реакції на поворот апаратної ручки. На жаль, складно створити тест, що оцінює що-небудь, крім простоти освоєння рішення новачками. Існує ряд методів для оцінки зручності продукту у використанні середнім користувачем або експертом, але такі методи віднімають багато часу і дають в (кращому випадку) не надто точні результати. [13, 14]

Існує безліч способів перевірити інтерфейс на реальних користувачів. Це - неформальні сеанси зворотного зв'язку, в яких пояснюються ідеї і вислуховуються міркування користувача, і більш суворі методи юзабіліті-тестування, в якому користувачі вирішують певний набір завдань. Кожен підхід має свої переваги. Більш неформальні методи можуть застосовуватися спонтанно і вимагають меншої підготовки. Мінус такого підходу в тому, що проектувальник часто виявляється винен в «підказках свідку», якщо пояснює в спонукає манері. В цілому, цей підхід прийнятний для технічної аудиторії, яка здатна аналізувати нечисленні малюнки, що представляють інтерфейс продукту.

### *1.2.1 Usability-тестування*

Usability-тестування - це експериментальний метод, побудований на інтерв'юванні користувачів за попередньо складеним планом і спрямований на з'ясування того, як користувачі використовують продукт.

Учасників тестування, також званих «респондентами», відбирають і наймають на основі спеціалізованої анкети, складеної так, щоб відсіяти людей, що не входять в цільову аудиторію користувачів продукту. У процесі кожного інтерв'ю модератор (фасилітатор) просить респондентів виконувати завдання, які вирішуються за допомогою тестованого продукту.

Учасниками такого тестування є звичайні користувачі з числа тих, хто працює (або приблизно буде працювати) з об'єктом тестування. Це відрізняє їх від експертних оцінок, де свій висновок з даного питання в тій чи іншій формі дають юзабіліті-фахівці.

Сенс usability-тестування полягає в спостереженні та аналізі проблем, з якими стикаються користувачі, працюючи з продуктом.

Метою тестування є поліпшення інтерфейсів на базі досвіду цільової аудиторії. При цьому створюється фінальна версія продукту, що дозволяє забезпечити найбільш ефективну взаємодію. [8]

Спостереження за користувачами дозволяє знайти способи поліпшення дизайну і функціональності продукту.

### *1.2.2 Оцінка продуктивності*

Оцінка продуктивності взаємодії користувачів з системою (Performance measurement) полягає в тому, що групу користувачів просять виконати певний набір завдань. При цьому вимірюють час, яке їм знадобиться, число помилок, яке вони зробили, а також при необхідності деякі інші параметри (частота звернення до довідкових матеріалів, число відвіданих сторінок на сайті і т. П.).

Під час проведення такої оцінки спілкування між організатором і респондентом повинно містити лише до пояснення завдань, які необхідно виконати. В інший час організатор повинен грати роль пасивного спостерігача і не втручатися в процес роботи користувача, якщо тільки той не виявився в ситуації, коли він не здатний самостійно вирішити проблему. Тільки при дотриманні цієї вимоги можна розраховувати на об'єктивність отриманих даних. [9]

Важливою частиною такого підходу є аналіз отриманої інформації. Необхідно вміти правильно інтерпретувати зібрані дані. Кількісні дані не зможуть показати, в чому конкретно полягає проблема, але вони добре підходять для визначення того, де зосереджені найбільш серйозні недоліки, або для порівняння різних інтерфейсів. Та й самі результати вимірювань можуть бути обумовлені різними факторами. Можливо, що помилка викликана не серйозними проблемами в системі, а просто неухважністю

конкретного користувача. Проте всі помилки необхідно розглядати більш детально, прагнучи визначити викликали їх причини.

Окрім характеристик, які можна об'єктивно виміряти, існує такий параметр, як задоволеність користувача від роботи з системою. Його також бажано враховувати. Для цього з респондентом потрібно провести невелику бесіду або попросити його заповнити спеціальну анкету після завершення виконання завдань. Інший варіант дій - спостерігати за поведінкою користувача під час роботи з веб сайтом і спробувати зрозуміти відчуття від взаємодії з непрямими ознаками і реакцій.

### *1.2.3 Фокус групи*

Застосування фокус-груп вважається фінансово не вигідним рішенням, так як в них мінімальне число учасників більше, ніж в інших дослідженнях. Крім того, для забезпечення необхідної точності результатів необхідно проводити кілька сеансів з різними респондентами.

Мета фокус-групи полягає в зборі якісних даних на основі ретельного обговорення учасниками конкретної проблеми (питання).

Метод фокус-груп дозволяє зрозуміти, що насправді люди думають і відчувають при роботі з проєктованим продуктом. У роботі кожної фокус-групи беруть участь від 4 до 12 осіб. Це - люди, попередньо відібрані відповідно до визначених критеріїв [18].

Обговорення проблеми (питання) в фокус-групі проходить в умовах, близьких до природних, що виникають при роботі з продуктом. Групове обговорення фокусується на певній тематиці. Питання до учасників і план проведення фокус-групи ретельно готують заздалегідь. Групову дискусію організовує професійний інтерв'юер. Він виконує функції модератора. Його завдання - створити в групі безпечну і комфортну атмосферу, спонукати учасників вільно висловлюватися по заданій темі і вислуховувати думку всіх.

В середньому участь у фокус-групі триває від 1 до 2 годин. Після завершення дискусії результати обговорення піддають ретельному аналізу, проводять узагальнення і систематизацію зібраної інформації.

Даний метод слід використовувати перш за все для того, щоб визначити, які функції користувачі хотіли б бачити в кінцевому продукті, в той час як питання, пов'язані з організацією інтерфейсу і особливостями його сприйняття користувачами, повинні розглядатися з застосуванням інших методик. В порівнянні з більшістю інших об'єктів, для дослідження ставлення до яких застосовуються фокус-групи, програмні продукти повинні не просто подобатися користувачам, а користувачам повинно бути легко та ефективно ними користуватися. Єдиним надійним способом перевірити це є спостереження за індивідуальним взаємодією користувача з розглянутою системою. [28-30] Фокус-групи найчастіше не мають на увазі таку можливість: зазвичай учасникам просто демонструють, як з нею працює хтось інший. При цьому враження від спостереження за чужими діями сильно відрізняються від вражень, які людина отримує від самостійного спілкування з продуктом.

Виходячи з цього, можна сказати, що в дослідженнях, пов'язаних з юзабіліті-характеристиками програмних продуктів, фокус-групи має сенс застосовувати лише на початкових етапах проектування, щоб виявити основні потреби і очікування потенційних користувачів. Застосування їх для оцінки існуючих або проєктованих інтерфейсів не ефективно і не дає позитивних результатів.

#### *1.2.4 Експертна оцінка*

На відміну від методик, що були описані раніше, учасниками експертної оцінки не є майбутні користувачі, а експерти в області юзабіліті і професійні дизайнери інтерфейсів. Вони досліджують продукт і намагаються виявити наявні, на їх погляд, проблеми[27].

До переваг цього методу можна додати те, що експерт здатний знайти майбутні проблеми на початкових стадіях розробки призначеного для користувача інтерфейсу.

До мінусів наступні пункти:

1. Оцінка експерта ґрунтується на досвіді, який був отриманий раніше.
2. Експертна оцінка - оцінка суб'єктивна.
3. Експертна оцінка повинна даватися фахівцем, що володіє як великим досвідом, так і високою кваліфікацією.

Експертна оцінка часто не може замінити собою повноцінне юзабіліті-тестування, але у неї є свої переваги. Існують різні формальні способи проведення такого роду оцінок. Нільсен виділяє вісім різних варіацій (наприклад, уявне переміщення по інтерфейсу, коли експерт продумує всі можливі шляхи руху користувача і шукає потенційно небезпечні місця). Навряд чи потрібно розглядати всі з них, зупинимося лише на методі евристичної оцінки, який отримав найбільшу популярність і поширення [31].

### *1.2.5 Евристична оцінка*

Основною особливістю даної методики (Heuristic Evaluation) є наявність списку певних юзабіліті-принципів (або евристик). Беруть участь фахівці по черзі досліджують продукт, виділяють існуючі, на їх погляд, недоліки і класифікують ці недоліки як порушення одного або кількох з цих принципів.

Список евристик заздалегідь визначається організаторами дослідження. Вони можуть варіюватися, але найбільш відомий набір з 10 правил, запропонований Я. Нільсеном. Вони сформульовані як універсальні для всіх програмних продуктів:

1. Видимість стану системи. Користувач завжди повинен бути поінформований про те, що відбувається в системі за допомогою явних засобів і протягом короткого часу.

2. Відповідність між системою і аудиторією. Система повинна спілкуватися з користувачем звичною для нього мовою.

3. Свобода дій користувача. Користувачі часто вибирають ту або іншу дію помилково, і у них повинна бути можливість без зайвих зусиль вивести систему з небажаного стану.

4. Послідовність і стандарти. У користувачів не повинні виникати сумніви, щодо визначення тих чи інших позначень або назв. Потрібно дотримуватися загальноприйнятих в даній області стандартів і домовленостей.

5. Запобігання помилок. «Краще займатися не дизайном повідомлень про помилки, а дизайном, який би запобіг виникненню помилок».



6. Розуміння краще запам'ятовування. Всі об'єкти, функції і дії в інтерфейсі повинні знаходитися перед очима користувача. Не потрібно змушувати його тримати в пам'яті ту інформацію, що може бути запам'ятовано системою. Інформація щодо користування конкретним елементом продукту завжди повинна бути доступна при першій необхідності.

7. Гнучкість і ефективність використання. У продукті повинні бути присутні функції, які не очевидні для новачків і не заважають їм, але дозволяють підвищити швидкість і ефективність роботи досвідчених користувачів. Потреби обох груп повинні бути задоволені.

8. Естетичний і мінімалістичний дизайн. Будь-яка зайва інформація знижує рівень наочності інтерфейсу і ступінь ефективності роботи з ним.

9. Визначення, діагностування та виправлення помилок. Повідомлення про помилках повинні бути зрозумілі користувачеві, чітко визначати причину виниклих проблем і пропонувати конкретний спосіб їх вирішення.

10. Довідка та документація. Вся довідкова інформація повинна бути доступна, орієнтована на потреби користувача, містити конкретні кроки і мати розумний обсяг.

Евристична оцінка, яку дає кожен конкретний учасник, в більшості залежить від його особистих характеристик і особливостей сприйняття, тому, хоча дану методику і можна використовувати навіть при наявності єдиного експерта, ефективність в такому випадку буде значно нижче, ніж при груповій оцінці. Одна людина ніколи не зможе виявити всі наявні в інтерфейсі проблеми. Я. Нільсен зазначає, що п'ять експертів часто в змозі виявити близько 75% недоліків, і це - досить хороший результат. Три людини - це мінімальне число учасників, при якому має сенс проводити оцінку [32,33].

Так як в ході евристичної оцінки учасники не використовують систему на пряму (не виконують ті завдання, які, ймовірно, будуть вирішувати з її допомогою користувачі), то таку оцінку можна застосовувати і для оцінки інтерфейсів на самих ранніх етапах розробки, коли вони існують лише на папері. Методика підходить для

виявлення проблем юзабіліті на різних стадіях - і для початкового проектування, і для доопрацювання вже реалізованих рішень

### *1.2.6 Макетування (прототипування)*

Починати реалізацію продукту, базуючись на неперевірених даних і припущеннях, досить небезпечно: адже втрати ресурсів у випадку необхідності виправлення помилок, які з'являються на стадії проектування, можуть бути дуже значними, тому має сенс починати юзабіліті-тестування якомога раніше. Для того щоб залучити в процес передбачуваних користувачів, можна використовувати прототипи кінцевої системи, створення яких не потребує значних витрат часу і коштів. Прототип неминуче відрізняється від готового продукту за своїми характеристиками. Виділяють два основних види прототипів в залежності від того, в чому полягає відмінність. Вертикальні прототипи - це коли урізається число функцій кінцевої системи, але ті з них, що присутні, повністю працездатні (в такому випадку говорять про тестування «в глибину»). У горизонтальному прототипі, навпаки, позначаються всі передбачувані функції продукту, але вони не реалізуються (тестується головним чином дизайн інтерфейсу). Згідно з іншим принципом поділу прототипів також можна виділити дві групи - прототипи високої точності (high-fidelity) і низькою (low-fidelity). [7] Перші відрізняються від других тим, що виглядають і поведуться приблизно так само, як і кінцевий продукт, і на перший погляд їх важко розрізнити. А ось прототип низької точності навіть не дуже просунутий користувач ніколи не переплутає з справжнім інтерфейсом.

### *1.2.7 Сегментація аудиторії користувачів*

Потреби користувачів можна розбити на частини шляхом сегментації користувальницької аудиторії. Розподіл на різні групи проводиться за допомогою виділення подібних ключових характеристик представників кожної групи. Існує велика безліч різних способів сегментації цільової аудиторії. Вибір способу залежить від цільової аудиторії, особливостей продукту, що розробляється і т.д. На перших етапах сегментування може виходити велика кількість груп, але в процесі визначення пріоритетів деякі групи об'єднуються. В результаті виходять 3-4 основні цільові

групи. Для того щоб краще усвідомити, які цілі переслідують користувачі, необхідно спочатку зрозуміти, хто вони. Щоб зробити користувачів більш конкретними і зрозумілими, слід описувати (персоніфікувати) типових представників кожної групи цільової аудиторії, формуючи репрезентативну вибірку. Це повинна бути вибірка, що відображає властивості генеральної сукупності, в яку входять всі можливі користувачі, на яких спрямована увага розробників продукту.

### *1.2.8 Закон Фіттса (Fitts's Law)*

Закон Фіттса описує емпіричну (засновану на спостереженнях, а не на чистій логіці) модель сенсорно-моторних реакцій людини, яка застосовується для прогнозування часу, потрібного користувачеві для того, щоб вказати курсором миші на об'єкт на екрані комп'ютерного пристрою.

Що стала популярною з поширенням графічних користувацьких інтерфейсів (англ. Graphical user interface, GUI), в останні роки ця модель набуває все більшої важливості в розробці GUI, що пояснюється широким поширенням саме цього типу інтерфейсів. Модель враховує такі фактори як відстань до елемента на моніторі і розмір цього елемента, що дозволяє обчислити, наскільки легко і ефективно користувач зможе вибрати потрібний йому об'єкт[29].

Ці знання стануть у пригоді, наприклад, при проектуванні призову до дії, покликаною збільшити коефіцієнт конверсії посадкової сторінки. Спочатку сформульований в 1954 році, закон Фіттса залишається актуальним і сьогодні, незважаючи на зростання популярності пристроїв з сенсорним управлінням.

### *1.2.9 Моделі GOMS*

Оцінка ефективності різних системи можна реалізувати за допомогою групи методів GOMS (з англ. Goals, Methods, Rules). Для того завдання, яке користувач виконує в системі (наприклад, «коригування інформації про учасника угоди»), дії розкладаються на елементарні дії: переміщення курсору миші, натискання на клавішу клавіатури і т.д. Існують таблиці з експериментально обчисленими значеннями часу для кожного такого дії. Використовуючи ці таблиці, аналітик обчислює конкретний час, який користувач витрачає на роботу з інтерфейсом системи.

Серед мінусів, на практиці зазначені значення можуть варіюватися в широких межах; розрахована на досвідчених користувачів; не враховує контекст використання; не враховує помилки; не враховує фізичні обмеження; суб'єктивна оцінка фахівцем.

Серед плюсів, добре підходить, коли не потрібне точне значення часу; є можливість швидко та легко порівняти 2 інтерфейси.

### *1.2.9 Моделі СМО*

Сучасні аналітичні моделі масового обслуговування дозволяють обчислювати не тільки середній, а й максимальний час перебування заявок в системі або мережі. Під максимальним часом перебування заявки в системі або мережі масового обслуговування мають на увазі час, перевищення якого можливе тільки для деякої попередньо заданої частини заявок. [17] Теорія масового обслуговування дозволяє побудувати достатньо прості моделі функціонування інформаційних систем і дослідити поведінку цих систем в широкому діапазоні змін параметрів. Кожна із сторінок сайта, який проектуємо може бути представлена в аналітичній моделі у вигляді СМО, яка моделює певну затримку(час перебування користувача на сторінці), а модель всього сайта може бути представлена у вигляді відкритої мережі масового обслуговування.

### *1.2.9 Система MASP + MeMo*

Система MASP + MeMo [14] моделює взаємодію з урахуванням контексту використання і завдань (хоча модель користувача окремо не виділяється, вона включена в контекст), В цілому підхід виглядає багатограним, особливо з урахуванням сучасного бурхливого розвитку методів штучного інтелекту. Однак слід зауважити, що обчислювальна складність моделей може виявитися занадто високою (якщо тільки веб-інтерфейси не створюються для відносно простих завдань або особливих контекстів використання), а адаптація моделей до мінливих тенденцій в проектуванні юзабіліті занадто витратна.

### **1.2.10 Приховані марківські моделі**

Прихований марківський процес є зручним засобом для моделювання динамічних процесів, через це його можна ефективно використовувати при прогнозуванні поведінки на сайтах. Ключова особливість даної моделі в тому, що стани, які лежать в основі моделі, не спостерігаються явно, а знаходяться через параметри, які будуть описані пізніше. Основною ідеєю даної моделі можна назвати спробу прогнозу спостереження, ґрунтуючись на тому, який важить спостерігається величина на попередньому періоді.

Основними складовими прихованого марківського процесу є наступні параметри:

- 1) послідовність станів процесу
- 2) послідовність спостережень
- 3) початковий вектор ймовірностей для стану
- 4) матриця перехідних ймовірностей для станів
- 5) матриця ймовірностей для спостережень

Завдання побудови адекватної моделі користувача полягає в тому, щоб модель, з одного боку, не була надмірною (в цьому випадку станів більше, ніж потрібно для задоволення критеріїв коректності), але, з іншого боку, відповідала реальним умовам. Стан користувача в загальному випадку не може бути виміряна безпосередньо, ми бачимо лише запити, які генеруються при знаходженні в деякому стані. Справедливо описувати цей процес за допомогою прихованої дискретної Марківської моделі порядку (пам'яті)  $n > 0$ . Завдання побудови прихованих Марківських моделей (ПММ) добре описана в [11]. Нехай стани користувачів відповідають станам марківської моделі, а алфавіт складається з функцій сервісу, які викликаються користувачами:

$S = \{S_1, \dots, S_n\}$ -множина станів користувача

$Q = \{Q_1, \dots, Q_n\}$ -можливі запити користувачів

Після сформування перехідних ймовірностей ми можемо сформулювати задачу для користувача, щоб вирішити проблему оптимального пошуку шляху.

Як навчальну множину можливо брати послідовність станів користувача. За допомогою алгоритму (в [13] показано, що для задач прогнозування він є найкращим для більшості застосувань), майбутній стан користувача прогнозується виходячи з минулого набору. Для збільшення точності прогнозування, як навчальної вибірки, можна брати не  $n$  попередніх станів, а вибірку з минулих станів на підставі одного із

заданих фільтрів («перетин» подій). Наприклад, фільтр може бути заданий так: «події в  $n$ -й день тижня». Але мінусом даного методу є складні обчислення та необхідність визначення попередніх результатів, що не завжди є можливим.

Позначимо через  $x_t$  стан мережі в момент  $t$ . На виході мережі в момент  $t$  з'являється видиме для спостереження значення.

$y_t \in O = \{o_1, o_2, \dots, o_N\}$ , де  $N$  — число можливих різних спостерігаємих значень на виході.

Нехай  $\pi_j$  - початкова ймовірність знаходження мережі в стані  $i$ , а  $a_{ij}$  - ймовірності переходу з мережі стану  $i$  в стан  $j$ .

Нехай на виході мережі спостерігається послідовність  $y_1, \dots, y_t$ . Тоді найбільш ймовірна послідовність станів мережі  $x_1, \dots, x_t$  для спостережуваної послідовності може бути визначена за допомогою наступних рекурентних співвідношень: [2]

$$V_{1,k} = P(y_1 | k) * \pi_k$$

$$V_{t,k} = \max_{x \in S} P(y_t | k) * a_{x,k} * V_{t-1,x}$$

$V_{t,k}$  - це ймовірність найбільш ймовірної послідовності станів, що відповідає першим  $t$  значень, які можна побачити, що завершується в стані  $k$ . Шлях Вітербі може бути знайдений за допомогою покажчиків, запам'ятовуючих, який стан  $x$  з'являлося в другому рівнянні. Нехай  $Ptr_{t,k}$  - функція, яка повертає значення  $x$ , використане для підрахунку  $V_{t,k}$ , якщо  $t > 1$ , або  $k$  якщо  $t = 1$ . Тоді

$$x_T = \underset{x \in S}{\operatorname{argmax}}(V_{T,x})$$

$$x_{t-1} = Ptr(x_t, t)$$

Для розв'язання завдання прогнозування стану користувача необхідно описати зміну стану в часі. Представимо зміни станів як символи марківського джерела з пам'яттю (порядку)  $n$ .

Після Аналізу актуальних варіантів для оцінки юзабіліті було визначено їх головний недолік, це відсутність можливості оцінки адаптації під різні групи користувачів(рис. 1.1), тому потрібно створити комплексний підхід із обраних методів

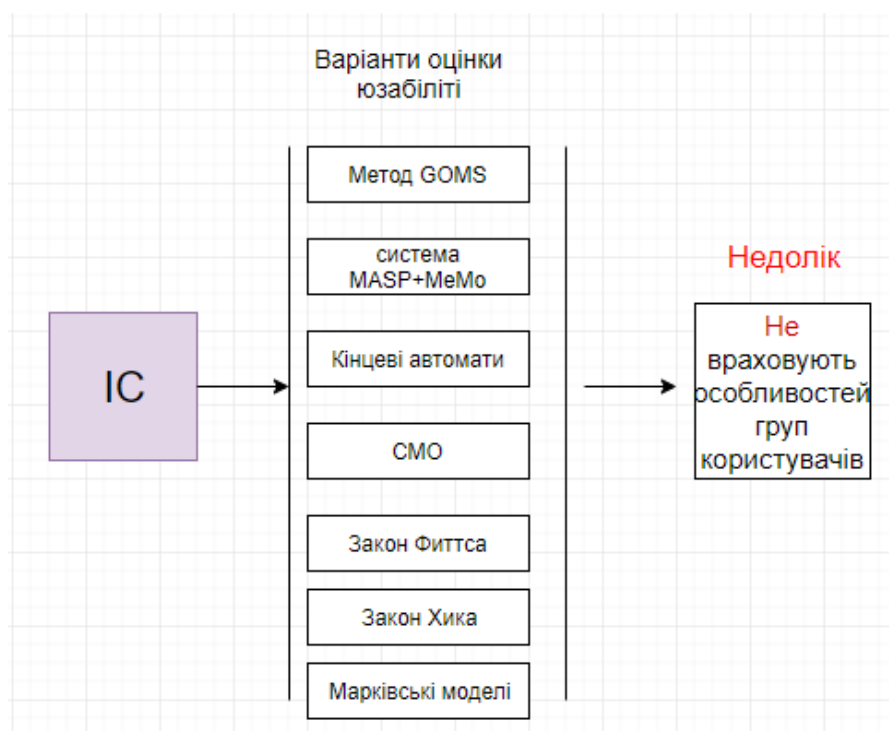


Рисунок 1.1-Результат аналізу методів оцінки юзабіліті

### 1.3 Аналіз програм-аналогів для створення прототипів веб-сайтів

Спершу ми обираємо програму в якій будемо створювати прототипи сайту. Програми для прототипування - це сполучна ланка між творчою ідеєю і кінцевим продуктом. Зовсім недавно в області UX / UI дизайну панував Photoshop, проте, проведене опитування показало, що цей найвідоміший графічний редактор став втрачати популярність серед розробників[4].

Велика проблема Photoshop полягає в тому, що це важкий інструмент, його власний формат PSD залишив дизайнерам управління великими файлами, а більшість його інструментів не використовуються для цифрових дизайнерів.

#### 1.3.1 Axure PR

Найбільший і найвідоміший додаток для прототипування, дозволяє створювати високоякісні прототипи сайтів і мобільних додатків. Володіє широким функціоналом з великим набором інструментів. Вимагає установки софта на ПК, онлайн-версія недоступна. Працює з Windows і macOS.

У Axure RP Pro можливе створення прототипів самого широкого діапазону - прототипів сайтів, схем технологічних процесів (і взагалі різних схем і діаграм), прототипів інтерфейсів програм тощо. (проте програма платна після 30 днів)

Можливості програми:

- Планування і прототипування для великих проектів, веб-сайтів і додатків
- Отримання зворотного зв'язку дає можливість вносити правки в дизайн ще до початку розробки.
- Інтерактивні прототипи і специфікації дозволяють всім, від замовників до розробників, працювати в одній команді.

Axure дозволяє спроектувати одну сторінку прототипу відразу для декількох розмірів екранів, за допомогою «адаптивних видів». Кнопка або поле введення, намальовані в одному адаптивному вигляді, з'являться у всіх інших. Зміни елементів і інтерактивність також успадковуються. [3] Таким чином, Axure дає можливість відносно швидко створювати адаптивні прототипи, а ще програма перевіряє збереження єдності між різними видами сторінок сайту.

### *1.3.2 Adobe Experience Design*

Програма входить в пакет Creative Suite від Adobe і синхронізується з багатьма сервісами з цього пакета. Головні особливості: швидкий перехід з прототипу в додаток, можливість моментально подивитися зміни в прототипі на мобільних пристроях. Головний мінус - десктопних додаток доступний тільки для Windows 10.

Зараз в Adobe XD плагіни доступні серед списку нових функцій, також багато плагінів Sketch вже перенесені в XD. Модулі були найбільшою перевагою Sketch, а тепер Adobe також отримав цю функцію.

Останніми роками Adobe XD розвинула екосистему символів. Тепер символи, скопійовані з одного документа і вставлені в прототипи, залишаються пов'язаними. Тепер, коли з вихідним файлом відбудуться зміни, у вас буде можливість застосувати оновлення в іншому файлі.

Програма також інтегрується з великою кількістю програм для спільної роботи і підвищення продуктивності, включаючи Slack, JIRA і Microsoft Teams



### 1.3.2 Sketch

Sketch дозволяє робити більш візуально привабливі прототипи. Це може бути дуже важливо при узгодженні прототипів з деякими керівниками. Буває так, що вони не вникають в усі деталі взаємодії і повноту функціоналу, оцінюючи прототип за принципом «подобається» - «не подобається». [6]

Програма надає багато інструментів для управління зовнішнім виглядом елементів і має доволі більш зручний інтерфейс для роботи з прототипом. Проте не дозволяє ефективно моделювати будь-яку взаємодія користувача з продуктом: спливаючі вікна, банерні каруселі, що випадають меню, динамічний контент на сторінці і т.д.

Цей інструмент створений ексклюзивно для Mac, тому він найкраще підійде для створення прототипів для додатків на iOS.

### 1.3.4 BALSAMIQ

Інструмент є доволі легким в користуванні, проте має обмежений функціонал для подальшої роботи, на етапах після прототипування. Має базові елементи для створення будь-яких інтерфейсів: мобільних додатків, сайтів, складних систем. Можна використати стандартні кнопки, іконки, списки, таблиці, меню, хлібні крихти.

У Balsamiq є можливість пов'язати посиланнями створеної сторінки і створити інтерактивний прототип. Це зручно використовувати для розуміння яка кнопка за що відповідає і як сторінки пов'язані один з одним. Пізніше можна перевірити їх дію, наприклад, в презентації. Презентацію зручно використовувати і для демонстрації готового прототипу замовнику. [8]

### 1.3.5 Figma

У Figma всі документи зберігаються в хмарі. Завдяки цьому в редакторі можна колективно працювати над макетами і відкривати їх за посиланням, без скачування.

Figma дозволяє зробити клікабельну версію сайту або мобільного додатку, створюючи безліч екранів. Можна протестувати і налаштувати будь-яку функцію: кнопки, спливаючі меню, анімовані і модальні вікна.

Всі екрани розміщені на одній сторінці. Перемикається між вікнами не потрібно. Ви створюєте ланцюг і бачите, як робота з інтерфейсом виглядає в реальності

Можна заходити в Figma через браузер або ж завантажити програму на комп'ютер. Вона підходить як для Windows, так і для Mac. У десктопної версії можна працювати офлайн, а коли з'являється доступ до інтернету, зміни синхронізуються.

Таблиця 1.1 – Порівняльний аналіз програм

Характеристика	<i>Axure PR</i>	<i>Adobe XD</i>	<i>Sketch</i>	BALSAMIQ	Figma
Онлайн платформа	-	-	-	-	+
безкоштовна	-(30 днів)	+	-	-(7 днів)	+
Інтерактивні прототипи	+	+	+	+	+
Зручна взаємодія з розробниками	-	+	+	-	+
Windows	+	+	-	+	+

Після аналізу було вирішено обрати Adobe XD оскільки програма має дуже зручний інтерфейс, є можливість створення інтерактивні прототипів і вбудована зручна взаємодія з розробниками веб-інтерфейсів.

### **Висновки до розділу 1**

Після аналізу наявних методів оцінки юзабіліті інформаційних систем. Було виявлено, що часто використовуваними методами на сьогоднішній день є

- Евристична оцінка
- Метод Фітса та Хіка
- Марківські процеси
- Сегментація аудиторії

Отже, було виявлено, що майже всі описані методи мають недолік це нечітка формалізація описаних результатів та неможливість оцінити готову інформаційну систему. Також ці методи не орієнтовані під різні потреби груп користувачів та оцінку адаптивності інтерфейсів.

## **2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ**

### **2.1 Мета та задачі**

Людино-машинний інтерфейс є каналом спілкування між користувачем і програмною системою [1]. Виділяється залежність між складністю інтерфейсу і набором знань, необхідних користувачеві для його використання. [2; 3]

Завдання адаптації інтерфейсу користувача інформаційних сервісів ставиться через необхідності залучення і утримання користувачів. З урахуванням цієї мети в завдання входить «спрощення» використання сервісу для певних груп користувачів.

Ергономічне проектування інформаційних систем і програмних засобів належить до найбільш перспективних напрямків досліджень в сучасній інженерній психології та психології праці[5; 6]. Зазвичай ергономічність програмного продукту (ПП) оцінюється на основі таких показників, як результативність, продуктивність і задоволеність користувача при роботі з ним

Метою роботи є дослідження та розробка моделей адаптивного управління людино-машинною взаємодією в інформаційних системах

Задачі:

- Розробити узагальнену модель інформаційної системи
- Провести оцінку рівня задоволеності користувачів інформаційною системою
- Розробити моделі для вибору варіантів людино-машинної взаємодії для реалізації адаптивного управління
- Провести апробацію моделей

### **2.2. Методи дослідження**

Задоволеність користувачів є важливою характеристикою якості програмних продуктів, зокрема інтернет-сайтів. Суттєвою перешкодою в емпіричних дослідженнях задоволеності є відсутність загальноприйнятих методичних засобів її оцінки.

Традиційно ергономічність програмного продукту (ПП) оцінюється на основі таких показників, як результативність, продуктивність і задоволеність користувача при роботі з ним (ISO ..., 1998). Однак якщо перші два критерії досить чітко оцінити і вони описані в спеціальній літературі, то розуміння джерела задоволеності користувача, детермінант і адекватних методів дослідження поки що немає. [11]

Задоволеність засобом або продуктом праці, тобто висока «суб'єктивна оцінка його якості», відображає особистісне ставлення до нього користувача. Воно формується на підставі попереднього досвіду взаємодії людини з певним об'єктом, в ході якого у нього неодноразово виникали стани комфорту і почуття задоволення.

Аналіз будемо здійснювати за існуючою методологією, яка описана в [25]. Для цього було проведене експертне опитування, до участі в якому залучалися особи різної кваліфікації - професійні розробники інтерфейсів, фахівці зі створення діагностичних тестів, а також основна цільова аудиторія, рядові користувачі з достатнім досвідом роботи з сайтом та без нього.

Виділення узагальнених компонентів в структурі приватних оцінок задоволеності інтернет-сайтом здійснювалося на основі факторного аналізу великого масиву даних, зібраного за результатами спеціально організованого інтерактивного опитування. Дана методологія на сьогоднішній день часто використовується та підтверджена числовими характеристиками, що згідно з коефіцієнтом кореляції Пірсона оцінки по кожному пункту опитувальника мають достовірні зв'язки не менше ніж з одним ( $r \geq 0.4$ ) і декількома іншими ( $r \geq 0.3$ ) пунктами. За критерієм Бартлетта матриця кореляцій значимо відрізняється від одиничної ( $\chi^2 = 10.688$ ,  $p < 0.001$ ), а значення статистики Кайзера-Мейера-Олкіна становить 0.956.

Виділені фактори добре диференціюються за складом пунктів опитувальника, які отримали максимальні навантаження. На рівні якісного аналізу відмінності між факторами легко проінтерпретувати як відносяться до різних аспектів задоволеності роботою з інтернет-сайтом. Так, найбільш потужний фактор 1 об'єднує пункти, що характеризують ступінь надійності, відсутності збоїв / труднощів і виникнення труднощів у роботі. Ми адаптували пункти опитувальника відносно нашої роботи з

сайтом ІТ курсів. Разом з експертами та маркетологами компанії було налаштовано ваги для кожного фактору. Було зроблено метою щоб усі фактори працювали з мінімальною різницею, але все одно була різниця в вагах відносно найважливішого компонента «Простота використання»

Таблиця 1.1- Фактори

Фактор	Приклади пунктів з максимальним факторним навантаженням
1-й фактор «Ефективність» Факторна вага: 0.9	Цей сайт часто працює некоректно 4) На сайті відсутні деякі з потрібних мені функцій 3) Сайт змушує робити багато зайвих дій (2) Під час роботи з сайтом виникають труднощі (1)
2-й фактор «Простота використання» Факторна вага: 1	Мені було легко зрозуміти, як працювати з сайтом (4) За зовнішнім виглядом сайту легко зрозуміти, як виконати будь яку дію (3) Всі дії, що виконуються на сайту для мене звичні, але їх важко знайти (2) Я не знайшов потрібних для мене дій (1)
3-й фактор «Корисність» Факторна вага: 0.8	Цей сайт потрібен мені для навчання (4) Сайт економить мій час (3) Надана сайтом інформація добре допомагає у вирішенні моїх завдань (2) Я не знайшов потрібних для мене дій (1)
4-й фактор «Емоційна привабливість» Факторна вага: 0.7	Цей сайт допомагає мені дізнатися корисну інформацію (4) Мені цікаво користуватися цим сайтом (3) Даний сайт занадто оригінальний (2) Я не пам'ятаю як виглядає сайт (1)

Людино-машинний інтерфейс є каналом спілкування між користувачем і програмною системою [1]. Виділяється [2; 3] залежність між складністю інтерфейсу і набором знань, необхідних користувачеві для його використання. Моделювання користувача - це процес побудови тим чи іншим чином інформаційної моделі користувача для адаптації інтерфейсу під нього. завдання

Завдання адаптації інтерфейсу користувача інформаційних сервісів виникає через необхідність залучення і утримання користувачів. З урахуванням цієї мети в завдання входить спрощення використання сервісу для користувача. Для

вирішення завдання адаптації інтерфейсу використовуються моделі по користувачам. Головною умовою для адаптації інтерфейсу є можливість його зміни в залежності від зовнішніх факторів [4], таких як дії користувача. Також для систем моделювання користувача важливим є велика кількість звернень, що допомагає будувати класифікацію моделей користувачів статистичними методами.

Через обробку великої кількості даних, одержуваних від користувача під час роботи з сервісом, можуть виникати такі проблеми: шум та помилкові дані. Також звичайні користувачі можуть не активно взаємодіяти з сервісами, які збирають статистичну інформацію.

З метою відокремлення необхідних відомостей використовуються статистичні методи, для того щоб передбачити зміну невідомого параметра від попередньо відомих параметрів. Це дає можливість здійснювати гіпотези щодо незнайомих параметрів моделі користувача відповідно до його дій. Також, можливо здійснювати теорії про майбутні дії користувача, для чого і існують статистичні моделі прогнозування [5], де невідомий параметр – майбутня дія користувача, наприклад, його місії, побажання щодо дій. З метою побудови моделі користувача з числа інших застосовуються відповідні методи:

- прямолінійні моделі;
- нейронні мережі;
- марківські моделі;
- кластеризація даних.

Дані способи зображені у роботах[7-9], в якому місці вони активно використовуються з метою адаптації інтерфейсів юзеру, коли потрібно показати рекомендовані новини, або завчасно завантажувати інтернет сторінки.

Після аналізу всіх методів було вирішено обрати для опису процесів сайту моделі Маркова, які значно переважають всі конкуруючі існуючі підходи.

Математичний апарат марківських моделей (ПММ) являє собою універсальний інструмент опису випадкових процесів, для роботи з якими не існує точних

математичних моделей, а їх властивості змінюються з плином часу відповідно до деяких статистичних законів.

Дуже часто появу випадкових подій описують у вигляді ймовірностей переходів з одного стану інформаційної системи в інший, так як при цьому вважається, що, перейшовши в один зі станів, система не повинна далі враховувати обставини того, як вона потрапила в цей стан.

Випадковий процес називається Марківським процесом (або процесом без післядії), якщо для кожного моменту часу  $t$  ймовірність будь-якого стану системи в майбутньому залежить тільки від її стану на даний момент і не залежить від того, як система потрапила в цей стан.

### *Опис марківських моделей*

Марківські моделі зараз широко використовуються для того щоб змоделювати поведінку користувача в навігації на сайті. Структура зв'язків посилань на сайтах може бути представлена як джерело мережі.

Якщо переглядати процес навігації користувачів Інтернету на сайті через Марківський ланцюг, то ми зможемо побудувати модель Маркова з сайту, з використанням переходів які здійснили користувачі на сайті. Ми можемо побудувати граф посилань, в якому сторінки - це вузли, а переходи між сторінками це зв'язки між вузлами [17]. Ми припускаємо, що більшість користувачів переходять за посиланнями на сторінки, які їх зацікавили. Переглядаючи ваги по посиланнях, з якими попередні користувачі неявно пов'язували свої переваги в наступній гіперпосиланням на кожній сторінці [8], можна використовувати граф посилань для розрахунку матриці ймовірностей переходу, що містить однокрокові ймовірності переходу між станами в моделі Маркова.

Для структури сайту, в якому з кожної сторінки можна потрапити на кожен, будується узагальнена напівмарківських модель, стану якої є сполученими, а граф переходів - компонентою сильної зв'язності. Граф станів представлений на рис. 2.  $S_0$  - перша або головна сторінка сайту, яка має 0-й рівень.  $S_{11} \dots S_{ij}$  - сторінки, які доступні для переходу з головною. Кількість рівнів вкладеності і кількість сторінок

різного рівня  $n$ . Таким чином, безліч станів утворюють  $\{S_{11}, S_{12}, \dots, S_{ij}, \dots\}$ . Вектор початкових ймовірностей  $(1, 0, \dots, 0)$ .

Атомарною дією користувача будемо називати дію користувача, необхідну для переходу від одного представлення призначеного для користувача інтерфейсу (екрану) до іншого. Залежно від реалізації інтерфейсу це може бути, наприклад, наведення і клацання миші на пункті меню в разі віконного інтерфейсу або вибір пункту меню з випадаючого меню-гамбургер.

Граф системи, що містить  $n$  вершин, можна поставити у відповідність матрицю  $n \times n$ , елементами якої є ймовірності переходів  $P_{ij}$  між вершинами графа, яка називається матрицею ймовірностей переходів. Елементи матриці задовольняють умовам:

$$0 \leq P_{ij} \leq 1 \quad (1.1)$$

$$\sum_{j=1}^n P_{ij} = 1 \quad (1.2)$$

Умова (1.1) - звичайна властивість ймовірностей, а умова (1.2) означає, що система  $S_i$  обов'язково або переходить з якогось стану  $S_i$  в інший стан, або залишається в стані  $S_i$ . Елементи  $P_{ij}$  матриці  $P$  значать ймовірності переходів в системі за один крок.

Зазвичай на графі, ймовірності переходу системи з одного стану в той же самий не відзначаються. При дослідженні конкретних систем зручно спочатку побудувати граф станів, потім визначити ймовірність переходів системи з одного стану в той же самий (виходячи з необхідної умови рівності одиниці суми елементів рядків матриці), а потім скласти матрицю переходів системи.

### 2.3 Методи реалізації математичної моделі

Оскільки марківські процеси пов'язані з введенням великої кількості даних та вирішенням задачі оптимізації, то було вирішено обрати для реалізації методу



електронні таблиці. Вони мають необмежені можливості збереження даних, дані зручно заносити до таблиць, а потім одразу використати у формулах.

З використанням електронних таблиць ми можемо працювати з даними різної складності, а також застосовувати вбудовані варіанти для пошуку рішень.

Серед переваг MS Excel ми виділяємо такі:

1. Можливість ефективного аналізу оброблених даних.
  - можна швидко обробити великі масиви даних і одержати результат в зручному та бажаному вигляді;
  - існує вбудований метод авто виправлення в формулах, що самостійно виділяє та корегує помилки які були здійснені при записі формули.
  - можливість використання вбудованих формул і функцій;
  - можна переглядати статистику на масиви даних;
  - досліджувати впливи різноманітних факторів на введені дані;
  - рішення типових задач оптимізації.
2. Величезний вибір засобів для редагування та відображення даних.
3. Зручний друк напряму електронних таблиць.
4. Над документами можна працювати командою зі спільним доступом.

Серед можливостей електронних таблиць ми можемо виділити:

- Вбудована довідка для роботи
- можливість запрограмувати власні макроси
- вибір типу даних для кожної комірки
- доступна робота з гіперпосиланнями.
- Зручно вносити формули
- Взаємодія з повним пакетом MS Office.

Засобом, яким ми будемо користуватися є надбудова «Пошук рішення» для аналізу даних в Excel. Використовуючи цей засіб ми можемо визначити за яких значень змінних формула в цільовій функції буде приймати потрібне оптимальне значення (мінімальне або максимальне). Для того, щоб розрахувати задане значення

можна застосовувати різноманітні математичні моделі та методи пошуку. Також усі дані можна одразу занести до обраної комірки.

З цієї надбудови ми будемо використовувати симплекс-метод за яким розрахуємо оптимальну функцію. Даний метод є методом цілеспрямованого перебору основних рішень задачі лінійного програмування. Він дозволяє за кінцеве число кроків знайти оптимальне рішення або встановити, що оптимальне рішення відсутнє.

Основний зміст симплексного методу полягає в наступному:

- Вказуємо спосіб знаходження оптимального опорного рішення
- Вказуємо спосіб переходу від одного опорного рішення до іншого, на якому значення цільової функції буде ближче до оптимального, тобто вказати спосіб покращення опорного рішення
- Задаємо критерії, які дозволять своєчасно припинити перебір опорних рішень на оптимальному рішенні або зробити висновок про відсутність оптимального рішення.

## **2.4. Планування робіт**

Детальний опис етапу планування робіт описано в додатку А

### **Висновки до розділу 2**

У розділі було виділено цілі проекту. Описано основні методи, які плануємо використовувати на наступних етапах. Зроблено диференціацію етапів необхідних для досягнення мети. Досліджено всі можливі ризики та розроблено план дій для кожного випадку окремо.

### 3. МОДЕЛЮВАННЯ СИСТЕМИ АДАПТИВНОГО УПРАВЛІННЯ

#### 3.1 Функціональне моделювання

Для візуального опису бізнес процесів було вирішено побудувати діаграму IDF0(рис. 3.1). В центральному блоці вказано процес який я описую: адаптивне управління людино машинною взаємодією в інформаційних системах.

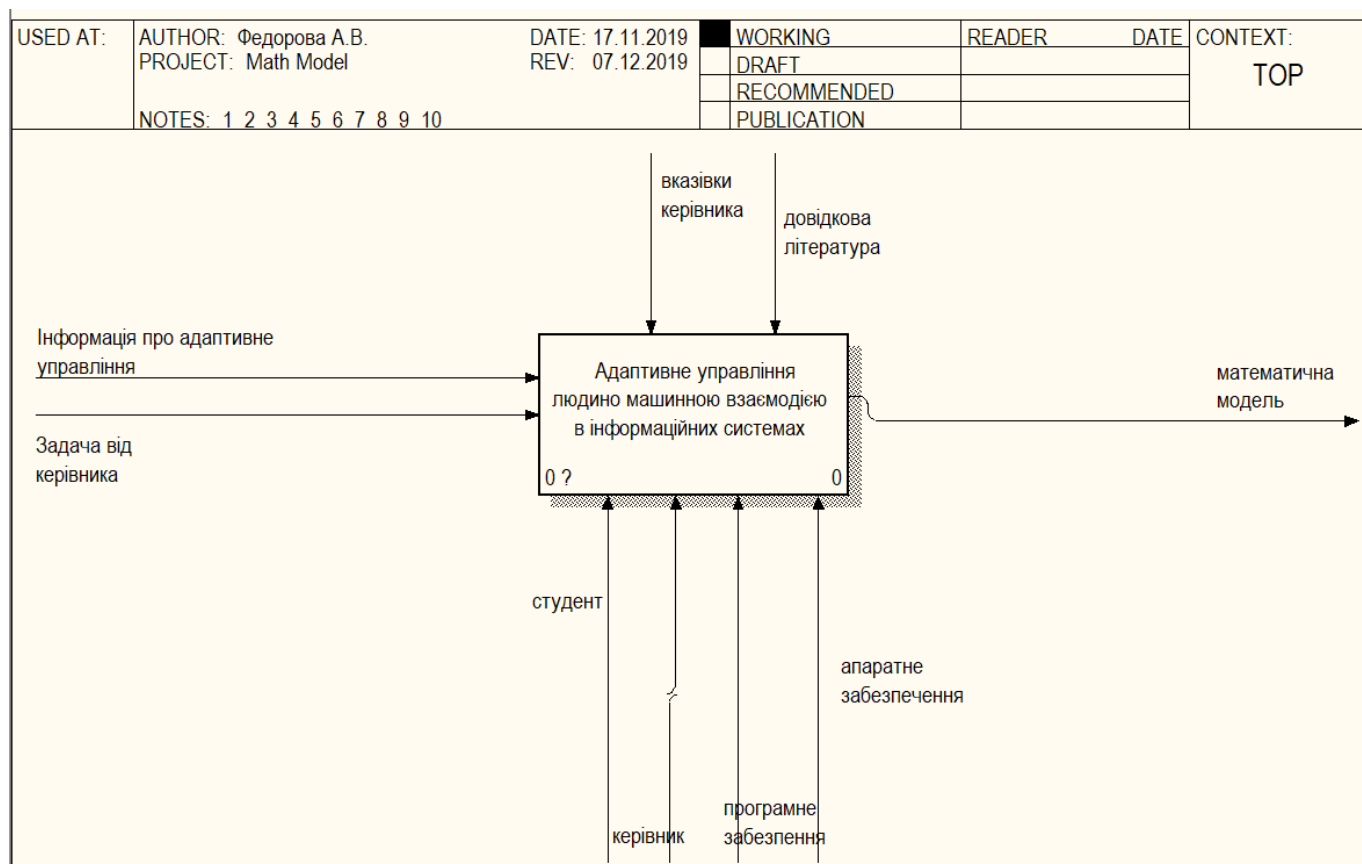


Рисунок 3.1 – Перший рівень діаграми

З правого боку на від діаграми приходять зібрана інформація про особливості адаптивного управління та поставлена задача від керівника. На виході ми отримуємо математичні моделі. Участь у процесі беруть студент, керівник та технічне забезпечення. На другому рівні у діаграмі виділено 4 пункти, які відображають процеси необхідні для реалізації (рис 3.2).

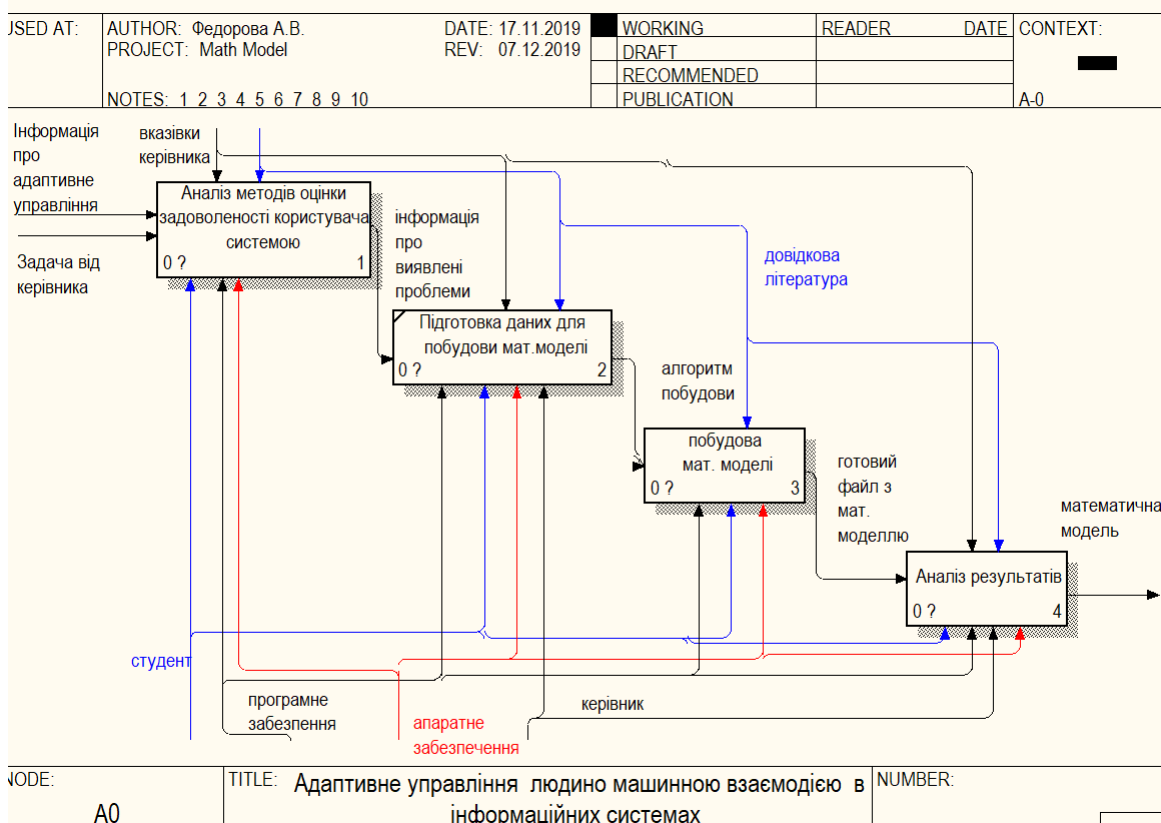


Рисунок 3.2 – Декомпований вигляд діаграми

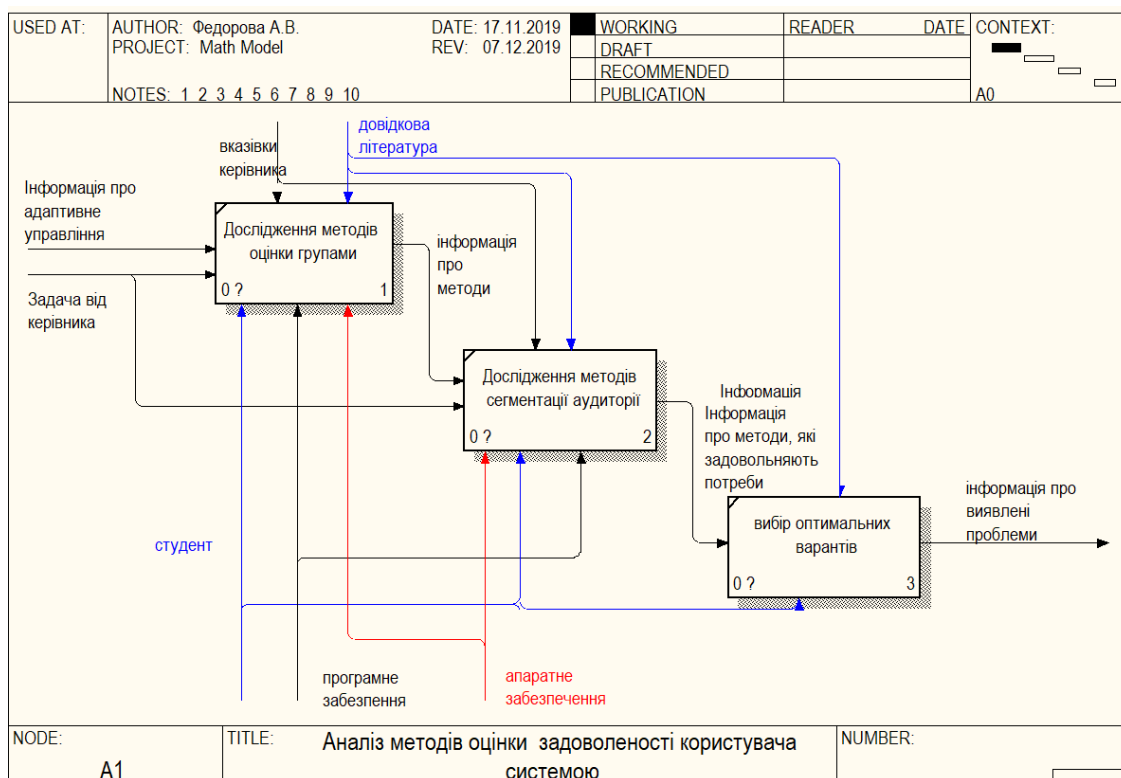


Рисунок 3.3 – Декомпозиція блоку «Аналіз методів оцінки задоволеності користувача системою»

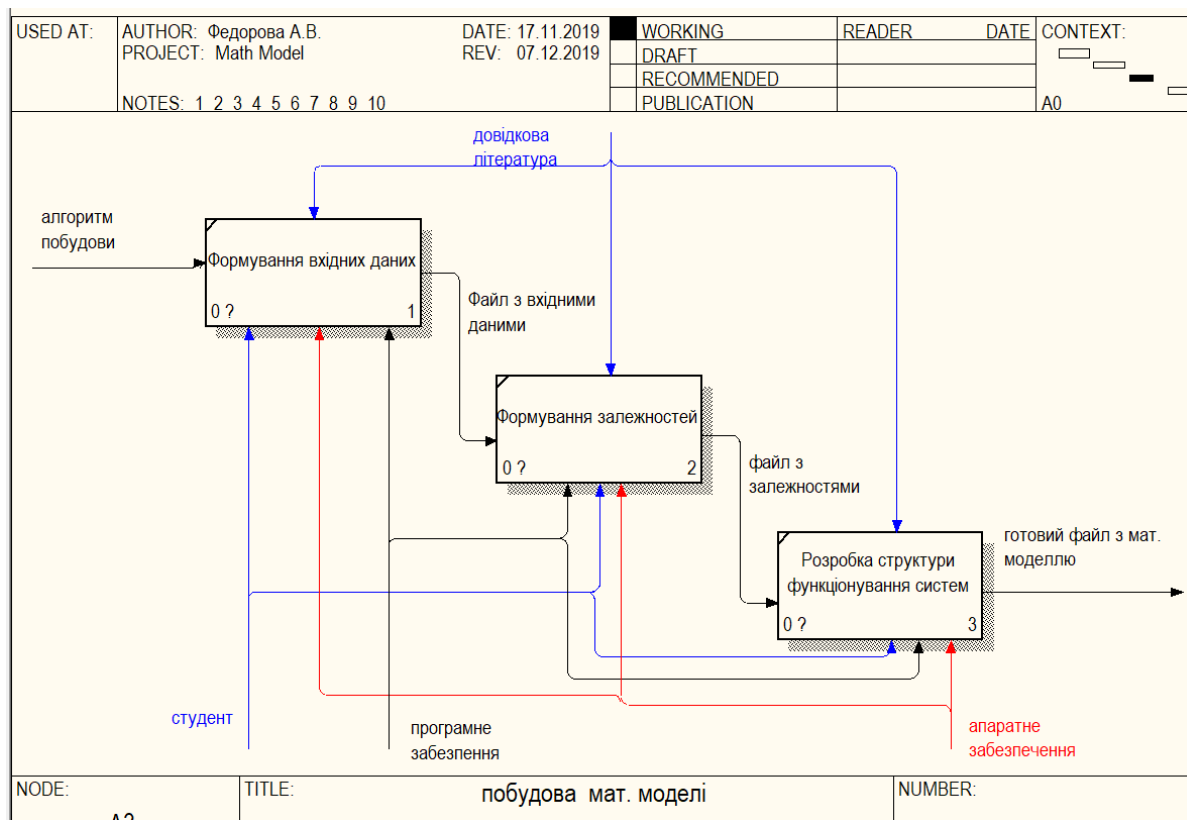


Рисунок 3.4 – Декомпозиція блоку «Побудова математичних моделей»

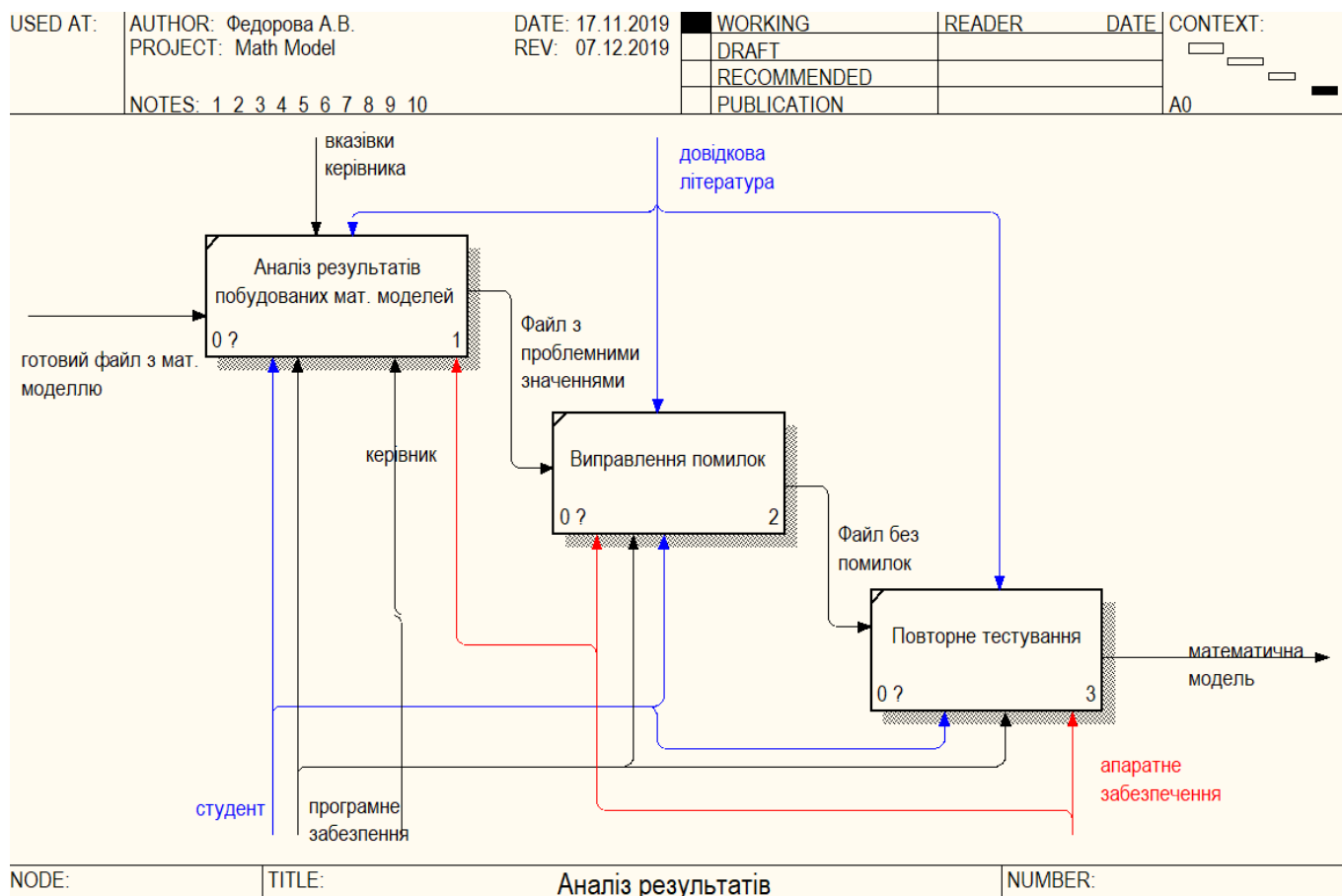


Рисунок 3.5 – Декомпозиція блоку «Аналіз результатів»

Всі наступні декомпозиції блоків третього рівня зображено на рис 3.3-3.5. На цьому декомпозицію було вирішено завершити, оскільки дані дії вже розкладені на достатньо прості пункти по моделюванню роботи.

### 3.2 Моделювання діаграми IDEF3

Для того, щоб відобразити процеси, які відбуваються під час роботи над створенням математичних моделей, ми створимо діаграми IDEF3. Ми виділяємо функціональні елементи, які відображають певні етапи процесу роботи над математичною моделлю (рис 3.6)

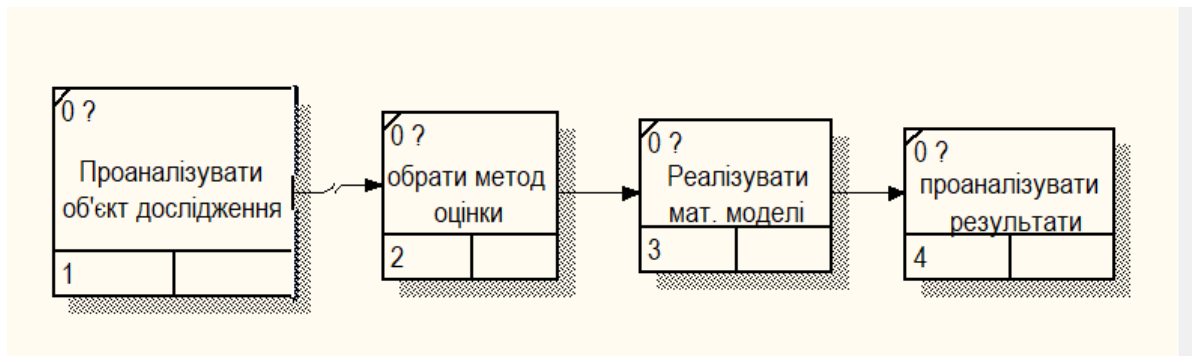


Рисунок 3.6- IDEF3 діаграма процесів

Наступним рівнем ми деталізуємо важливі процеси для більш детального розгляду. На рис. 3.7 зображено декомпований процес «Проаналізувати об'єкт дослідження»

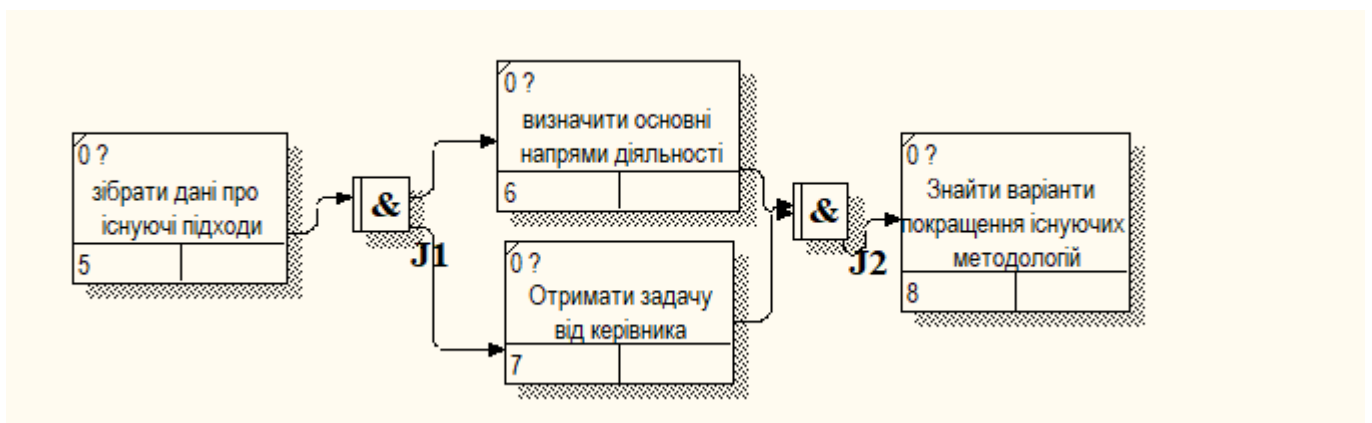


Рисунок 3.7- IDEF3 діаграма процесів

На рис 3.8 зображено детальний зв'язок процесів, які входять в блок «Реалізувати математичні моделі»

USED AT:	AUTHOR: Федорова АВ PROJECT: IDEF3_Math_model	DATE: 08.12.2019 REV: 08.12.2019	WORKING	READER	DATE	CONTEXT:
			DRAFT			<input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
			RECOMMENDED			
	NOTES: 1 2 3 4 5 6 7 8 9 10		PUBLICATION			1

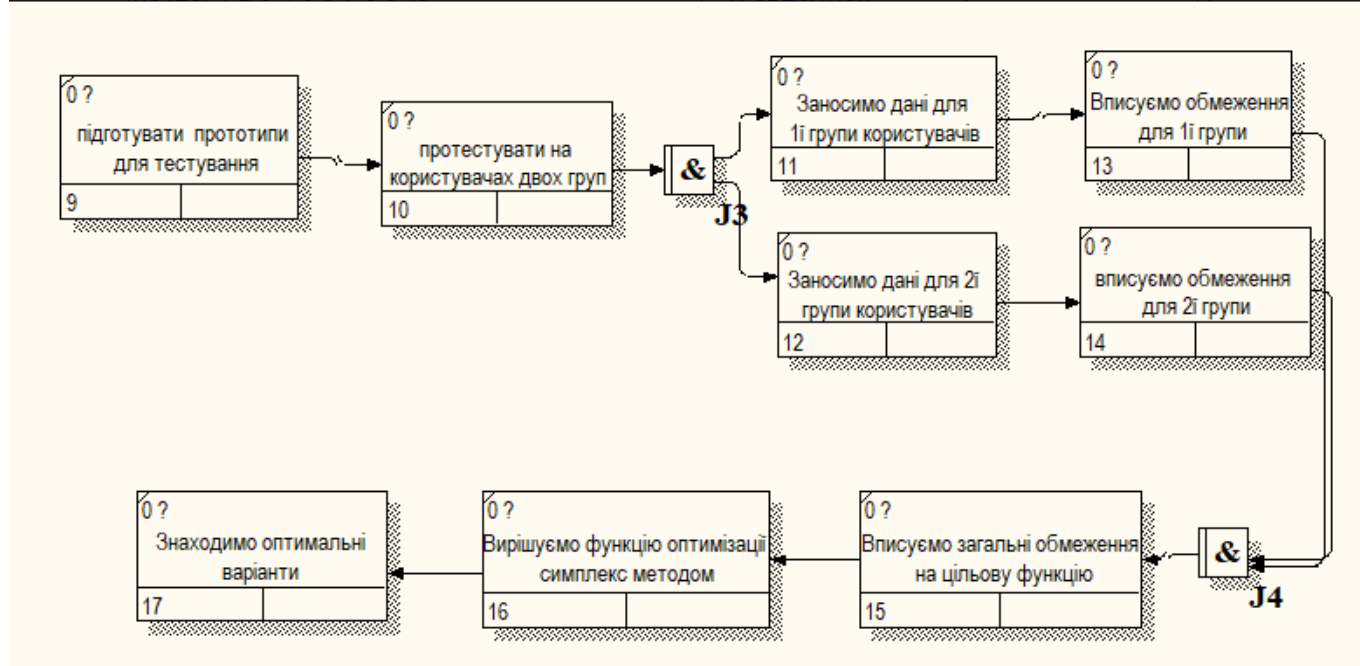


Рисунок 3.8- IDEF3 діаграма процесів для блока «Реалізувати математичні моделі»

### 3.3 Розробка графу системи

При роботі зі складними системами існує необхідність для створення альтернативних шляхів поведінки системи у випадках роботи з різними групами користувачів (рис 3.9).

Для реального використання методики адаптації сервісів під різні групи користувачів, потрібен чисельний критерій поліпшення ступеня адаптованості сервісу. Кількість часу, що витрачається користувачем на вирішення завдань, може залежати від багатьох факторів, в залежності від того, наскільки зрозумілий інтерфейс користувача і наскільки абонент знайомий з термінологією. Однак в загальному випадку кількість часу для вирішення завдання пропорційно кількості атомарних дій абонента, що дорівнює шляху в кількості ребер від точки входу до листа, який містить відповідні функції, які вирішують певну задачу абонента. Для спрощення схеми дослідження системи, можна вважати користувацький інтерфейс графом (рис 3.10). Складність функції інтерфейсу буде глибиною графа, на якому знаходиться ця функція.

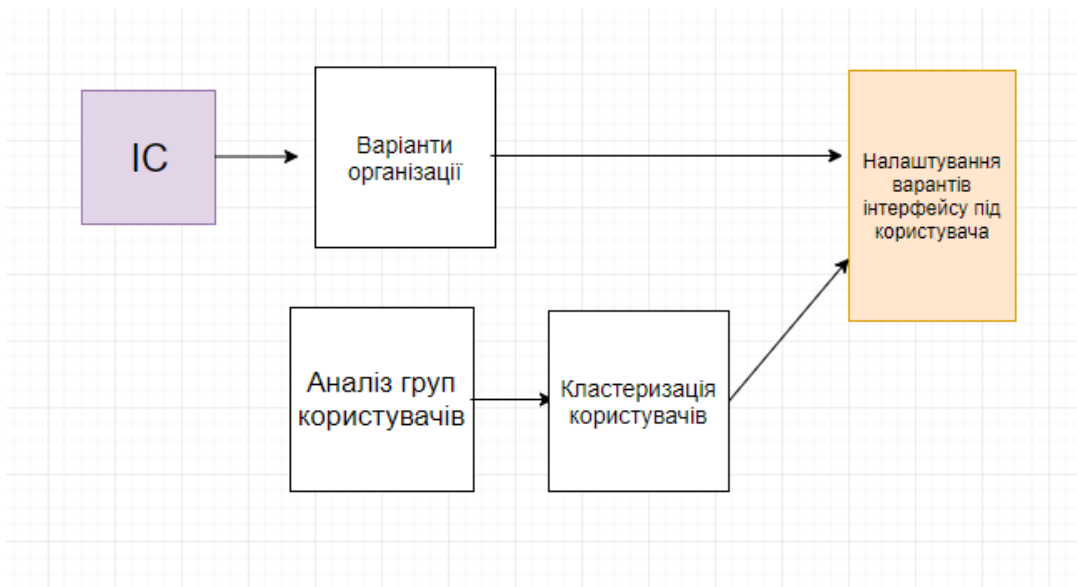


Рисунок 3.9- Схема аналізу системи для реалізації математичних моделей

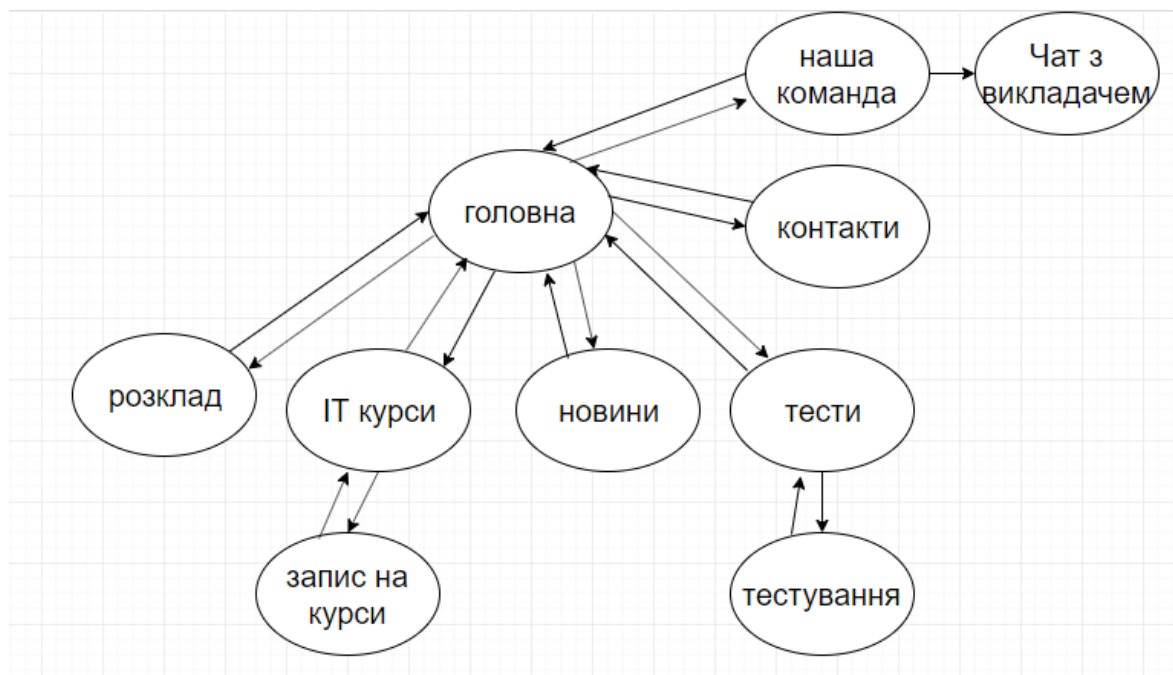


Рисунок 3.10- Приклад графу для сайту ІТ курсів

Глибина вузла  $t$  дерева - кількість ребер на шляху від вузла  $t$  до цільової дії. Будемо позначати глибину як  $d(t)$ . Нехай задано безліч користувачів  $U$ , для якого будемо порівнювати ступінь адаптованості інтерфейсів. Нехай є два різні описи інтерфейсів  $I$  та  $I_1$  - опис інтерфейсу сервісу у вигляді графа. При підрахуванні складності інтерфейса за рахунок дій, які потрібно виконати для досягнення цільової дії ми отримаємо дві різні складності  $M(I)$  та  $M(I_1)$ . Тоді можна отримати умову 3.1



$$M(I) - M(I_1) > 0 \quad (3.1)$$

За такої умови ми отримуємо, що інтерфейс  $I_1$  краще адаптований за інтерфейс  $I$ . Це рішення будемо використовувати під час створення альтернативних варіантів прототипів. Які будуть тестуватися групами цільової аудиторії. Оскільки даний варіант при самостійному використанні не вирішує проблеми різних потреб побудови інтерфейсу для різних груп користувачів.

### **Висновки до розділу 3**

У третьому розділі ми змоделювали етапи роботи за допомогою діаграм IDEF0 та IDEF3. Детальніше розглянули процеси, які потрібні реалізації математичних моделей.

Визначили, що розв'язувати задачу оптимізації цільової функції ми будемо вирішувати для двох різних груп користувачів з врахуванням альтернативних варіантів, які містять різні глибини доступу до цільової дії.

## 4. РЕАЛІЗАЦІЯ МАТЕМАТИЧНОЇ МОДЕЛІ

### 4.1 Вибір цільових груп користувачів

Реалізувати математичну модель та робити апробацію даних ми будемо на прикладі сайту ІТ курсів. Тому спершу для того щоб сформувати цільову аудиторію, яка буде тестувати варіанти наших дизайнів. За допомогою онлайн сервісів ми визначили основних відвідувачів. На рис 4.1 є схема розподілу користувачів за віком. Найбільші групи це люди віком 25-34 та 35-44.

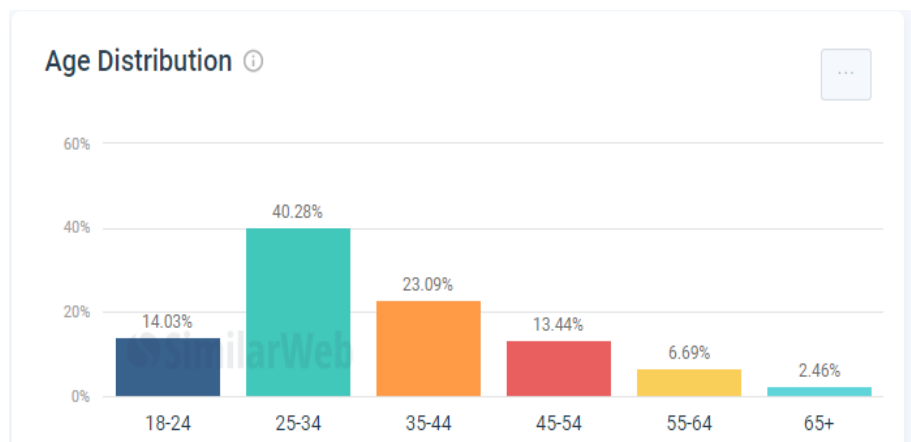


Рисунок 4.1- Графік розподілу вікових груп користувачів

Під час роботи будь-якого сайту постає питання про необхідність адаптації сайту на різні мови, але в нашому випадку, основні користувачі сайту українці, це видно з рис 4.2, тому такої необхідності немає.

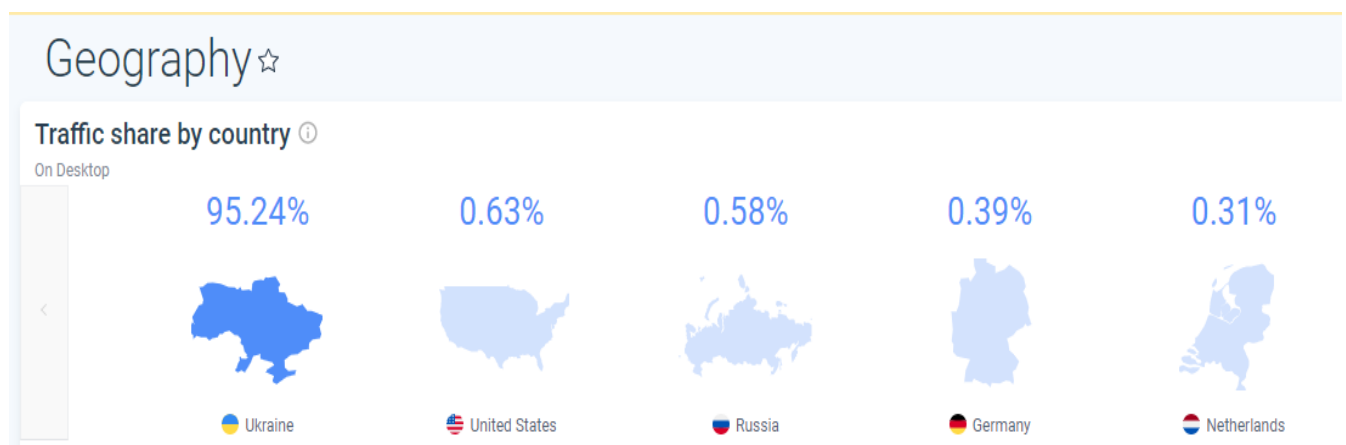


Рисунок 4.2- Розподіл користувачів за країнами

## 4.2 Факторний аналіз

Наступним етапом використовуємо факторний аналіз, методику якого описано в розділі 2.2. Обравши групу людей відповідно до попередньо описаних характеристик ми тестуємо наш існуючий сайт, за чотирма характеристиками, які зображені на рис 4.3(а).

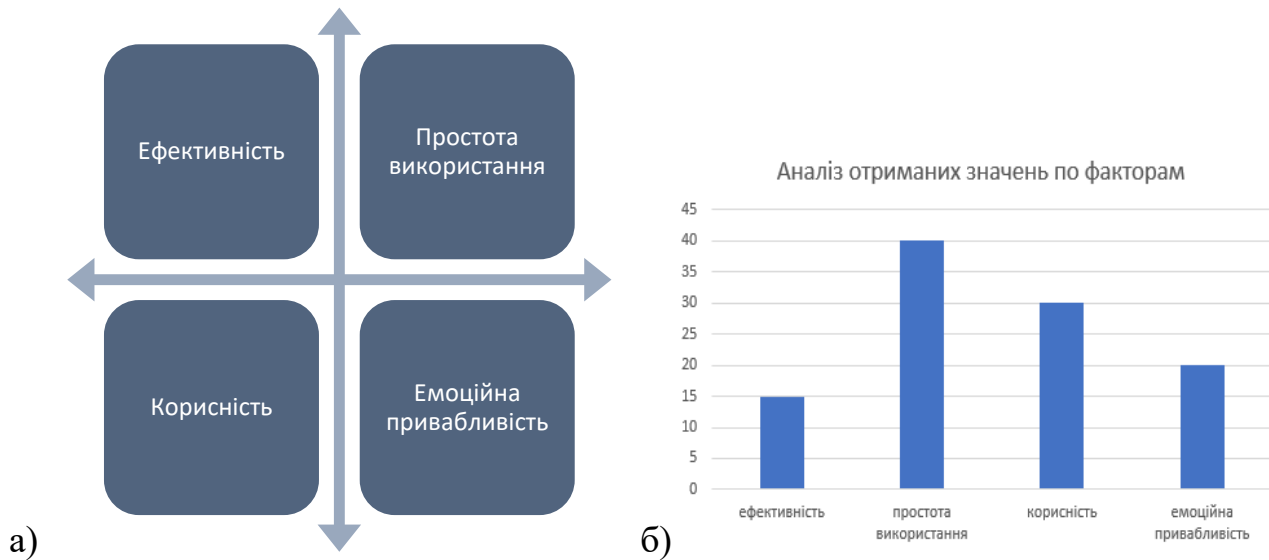


Рисунок 4.3- а) Фактори б) Отримані оцінки після факторного аналізу

Графік(рис. 4.3б) було побудовано з даних, які зібрані з 20 чоловік. Найнижчі числові характеристики отримав фактор «Ефективність», серед недоліків користувачі відмітили наявність зайвих дій, некоректну роботу сайту, застарілий дизайн інтерфейсу. Вже згруповані дані користувачів наведено в табл. 4.1.

Таблиця 4.1-Результати отриманих даних від користувачів

$T_0$	$K_0$	$T_0$	$K_0$
10	8.1	30	7.5
21	6.8	62	5.2
15	5.7	70	6.2
18	6.5	68	6.1
13	8.2	75	5.4
22	5.2	15	8.1
60	7.2	20	8.3
5	10.2	60	5
15	9.8	85	4.8
17	9	90	4.6

Після отримання даних нам потрібно кластеризувати дані. Кластеризація - це розбиття множини об'єктів на деякі однорідні підмножини (кластери), параметри яких спочатку невідомі. Для конкретного завдання кількість кластерів може бути довільним або фіксованим.

Для цього будемо використовувати середовище MATLAB. MATLAB в порівнянні з традиційними мовами програмування (C / C ++, Java, Pascal, FORTRAN) дозволяє на порядок скоротити час вирішення типових завдань і значно спрощує розв'язання типових алгоритмів, за рахунок вбудованої БД рішень.

Ми використаємо алгоритм Fuzzy c-means. Кластери є нечіткими множинами і кожна точка належить різним кластерам з різним ступенем приналежності. Точка відноситься до того чи іншого кластеру за критерієм максимуму приналежності даного кластеру. Цей метод дозволить нам розбити множину на певну кількість груп, таким чином ми отримаємо кількість груп користувачів, для яких потрібно робити різні версії адаптивного інтерфейсу.

Метою яку вирішує FCM-алгоритм, є створення автоматичної класифікації множини об'єктів, які визначаються векторами характеристик в просторі характеристик. Тобто, цей алгоритм визначить кластери і зможе класифікувати об'єкти. Кластери будуть представлені нечіткими множинами, але кордони між групами кластерів також будуть нечіткими.

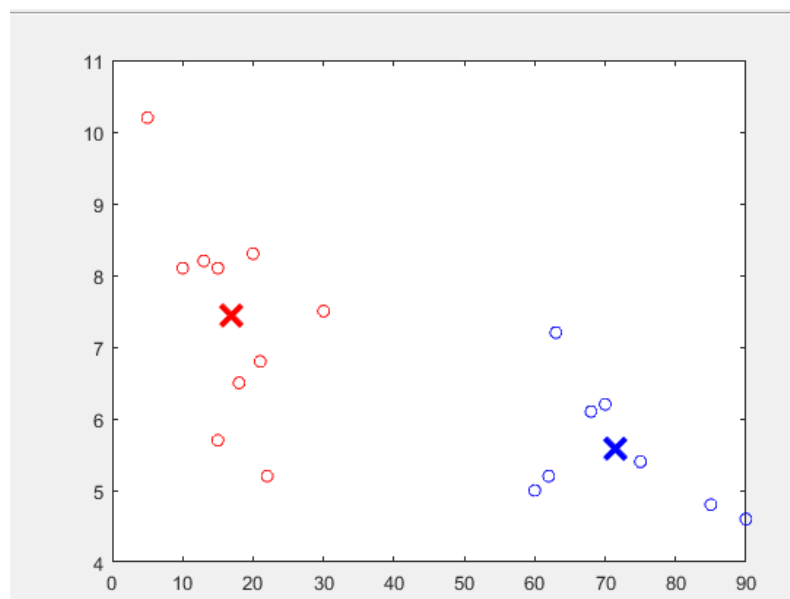


Рисунок 4.4- Кластеризовані дані

Після отримання результатів (рис. 4.4) ми можемо розділити наших користувачів на 2 окремі групи, відносно яких ми будемо аналізувати створений адаптивний дизайн. Код реалізації в MATLAB наведено в додатку Б.

### 4.3 Розробка підходу до розв'язання задачі адаптивного управління

Будемо використовувати граф переходів для того, щоб створити оптимізаційну модель. Головною метою ми ставимо забезпечити максимальну ймовірність потрапляння в заданий стан, який може відповідати події «Запис на курс», наприклад. Всі вершини графа потрібно пронумерувати натуральними числами.

$l$ -нумерація вершин

$d$  - поглинаючі вершини марківського процесу (тобто ті, які мають входи, але не мають виходів).

Для наших вершин ми описуємо вектор початкових ймовірностей, який власне описує ймовірність потрапляння системи на початкових станах у відповідну вершину на графі подій.  $y = (y_{l+1}, y_{l+2}, \dots, x_n), \sum_{i=l+1}^n y_i = 1$

Наступним кроком нам потрібно описати такі позначення:

$P_{ij}^{lm}$  - ймовірність того, що користувач  $l$  перейде з вершини  $i$  в вершину  $j$  якщо використовується  $m$ -тий спосіб реалізації ІС при стані  $S_i$ .

$n$  - кількість вершин, які описуємо.

$T_i^{lm}$  - математичне очікування величини терміну перебування користувача у вершині  $i$ , якщо використовується  $m$ -тий спосіб реалізації.

$T_0$  - часове обмеження на роботи з ІС

$K_i^m$  - математичне очікування значення оцінки користувача  $l$  рівня незадоволеності (дискомфорту) при перебуванні у вершині, якщо використовується  $m$ -тий спосіб реалізації.

$K_0$  - обмеження на оцінку рівня незадоволеності

$x_i^m$  - змінна яка буде описувати вибір рішення

$x_i^m > 0$  тоді, коли  $i$ -ої вершини обрано  $m$ -тий спосіб реалізації ІС, а рівне 0 в протилежному випадку.

$M_i$ - множина допустимих рішень в  $m$ -тому способі реалізації;

$N$  -загальна кількість вершин;

$\delta_i^m$ -булева змінна (0 або 1)

$W$ -велике обмежувальне число

$a, b, \dots, c$ , пов'язані стани, в яких будуть прийматися однакові рішення

Отже, тепер ми можемо описати формальний вигляд задачі

Для заданого користувача  $l \in \{0, 1, 2, \dots, l_0\}$ :

Виділяємо цільову функцію(формула 4.1)

$$\sum_{i=l+1}^N \sum_{m \in M_i} P_{ij}^{lm} x_i^m \rightarrow \max \quad (4.1)$$

$$\left\{ \begin{array}{l} \sum_{i=l+1}^N \sum_{m \in M_i} T_i^{lm} x_i^m \leq T_0 \end{array} \right. \quad (4.2)$$

$$\left\{ \begin{array}{l} \sum_{i=l+1}^N \sum_{m \in M_i} K_i^m x_i^m \leq K_0 \end{array} \right. \quad (4.3)$$

$$\left\{ \begin{array}{l} \sum_{m \in M_i} x_j^m - \sum_{i=l+1}^N \sum_{m \in M_i} K_i^m x_i^m = y_j, j = \overline{d+1, N} \end{array} \right. \quad (4.4)$$

$$\left\{ \begin{array}{l} x_j^m \geq 0, j = \overline{l+1, N}, m \in M_j \end{array} \right. \quad (4.5)$$

$$\left\{ \begin{array}{l} \sum_{m \in M_j} \delta_i^m = 1 \end{array} \right. \quad (4.6)$$

$$\left\{ \begin{array}{l} x_j^m - W \delta_i^m \leq 0, j = \overline{d+1, N}, m \in M_j \end{array} \right. \quad (4.7)$$

$$\left\{ \begin{array}{l} \delta_a^m = \delta_b^m = \dots = \delta_c^m, m \in M_j \end{array} \right. \quad (4.8)$$

Умови (4.2) та (4.3) нам дозволять знайти єдине розв'язання в вершині, де допускається тільки один варіант виконання операції.

Оскільки в ІС в різних режимах роботи кожен стан можливо реалізувати лише одним варіантом, а змінюватися спосіб буде тільки при переході в інший режим, ми будемо будувати стратегію, та введемо ці обмеження на кожную вершину.

Умови 4.4-4.5 обмежують варіанти коли кілька вершин мають однакові значення.

Умови 4.6-4.8 будуть обмежувати вибір ідентичних рішень в залежних станах системи

#### 4.4 Створення прототипів для груп користувачів

Сьогодні існує велика кількість програм з різним функціоналом, які дозволяють нам створювати одразу інтерактивні інтерфейси, для того що протестувати його на цільових групах. Для нашого випадку в попередньому розділі ми обрали Adobe XD (рис. 4.6).

Ми для себе вже визначили що користувачі нашого сайту ІТ курсів поділяють на дві групи умовно «досвідчені» та «новачки».

Відповідно до цього ми будемо реалізовувати по 2 варіанти прототипів на кожну групу. Під час створення ми орієнтуємося не на дизайн, а на структурну складову, для того щоб користувачі самостійно визначили, який з варіантів є найбільш зручним для них. Підтримуючи висунуте раніше припущення, що складність інтерфейсу залежить від глиби вузла на графі з цільовою дією, ми створимо інтерфейси різні за глибиною та структурою.

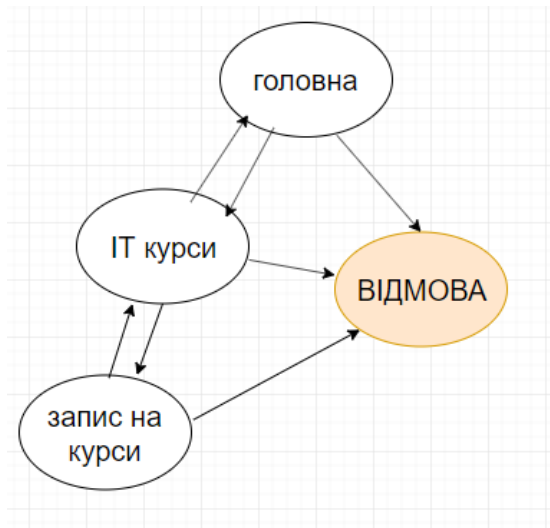


Рисунок 4.5- Граф подій який будемо реалізовувати на прототипах.

Спершу створює прототипи для 1ї категорії. Заплановано використати різні можливості для переходів між сторінками. Детально розглянуті всі прототипи буде описано у додатку В.

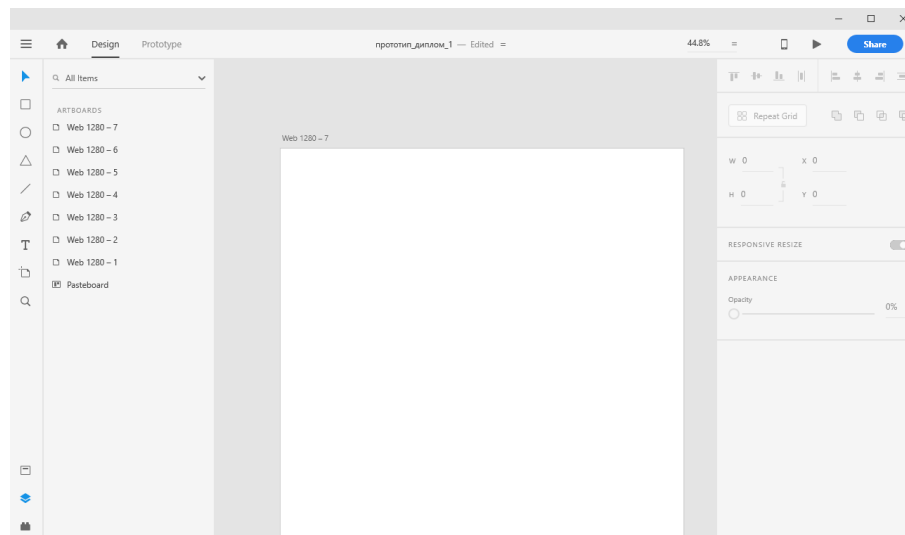


Рисунок 4.6- Вікно програми Adobe XD.

1й варіант прототипу зображено в додатку В

*2й варіант*

На 1 сторінці(головна) буде відображатися інформація про компанію, перехід на сторінку курсів можна здійснити після натискання на пункт меню «Курси» (рис. 4.7) Такий варіант структури було обрано для того, щоб оцінити наскільки досвідченим користувачам потрібна перша сторінка з не зовсім бажаною інформацією, та чи будуть вони вважати перехід з такої сторінки на сторінку «Курси» зайвим кроком. А в іншому варіанті прототипу ми навпаки скоротили кількість кліків та сторінка з курсами буде знаходитися одразу «Головній».

На рис. 4.7 можна помітити що ми користуємося вбудованою сіткою Bootstrap, яка є досить зручним інструментом при проектуванні інтерфейсів різних розмірів.

Також ми обрали великий розмір екрана на 1280, щоб в майбутньому була можливість створити різні версії адаптивного інтерфейсу під різні девайси.



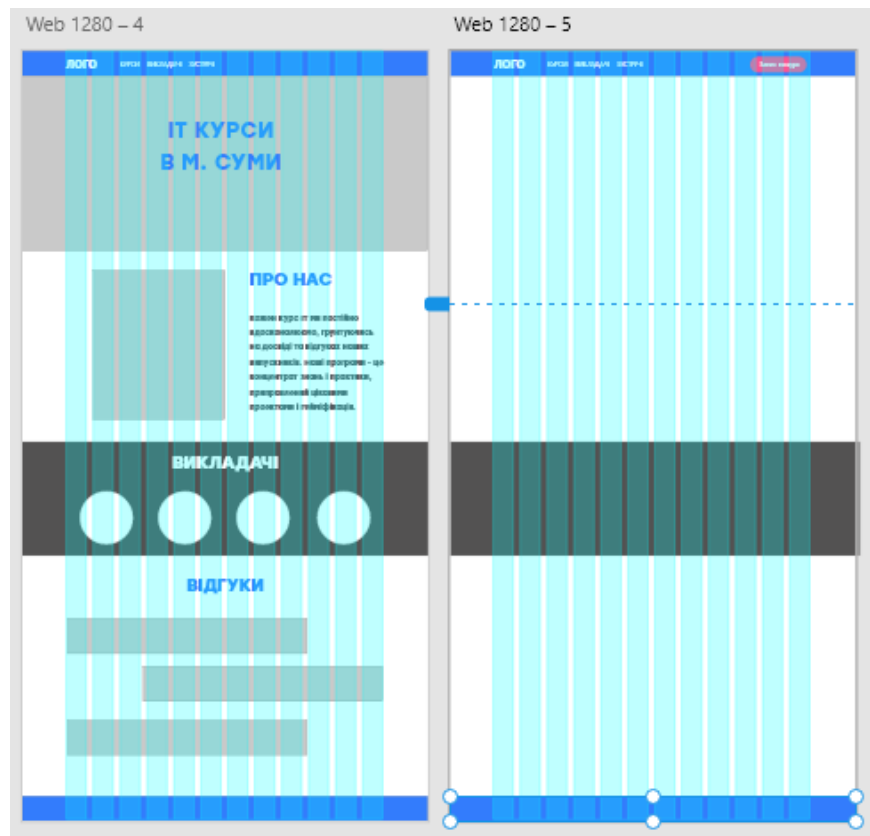


Рисунок 4.7- Прототип «Головної» сторінки ІТ курсів. Варіант 2

Наступним кроком, створюємо інші сторінки для цього прототипу (рис 4.7)

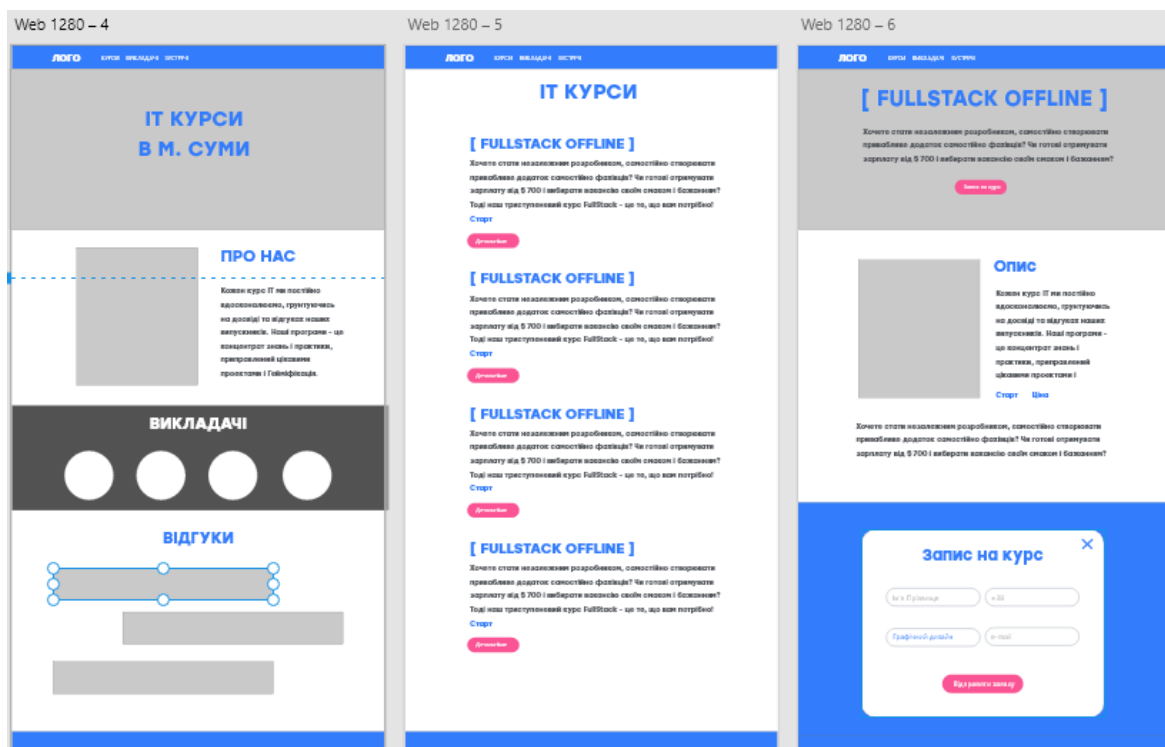


Рисунок 4.8- Прототип «Головної», «Курси», «Запис на курс» сторінки ІТ курсів. Варіант 2

Після реалізації всіх сторінок нам потрібно зробити інтерактивний прототип, який будемо використовувати при тестування. Нижче зображено приклад для переходу після натискання кнопки «Детальніше»(рис. 4.9).

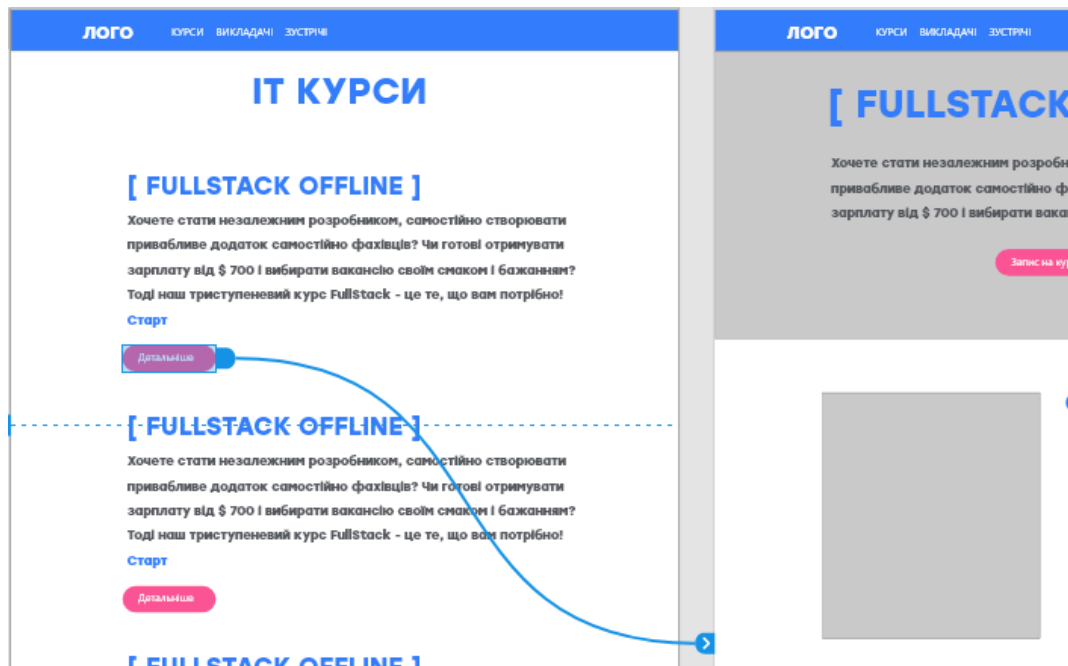


Рисунок 4.9- Реалізація зв'язків між сторінками сайту ІТ курсів. Варіант 2

Тепер продемонструємо приклад структури прототипу для 2ї категорії

1й варіант прототипу зображено в додатку В

*2й варіант для 2ї категорії користувачів*

Тут було вирішено також застосувати різні підходи та обрати різні види структури. На рис. 4.10 зображено «Головну» сторінку та одразу, на першому екрані користувачі зможуть бачити весь список наявних ІТ курсів. І по натисканню на один з них буде здійснюватися перехід на детальну сторінку курсу. Такий варіант структури сторінки є не досить типовим, тому ми вирішили перевірити чи зручно для користувачів «новачків» переглядати такі не звичні списки. Чи можливо вони оберуть інший варіант дизайну інформації про курси, який буде містити більше опису.



Рисунок 4.10- Прототип «Головної» сторінки ІТ курсів. Варіант 2 для 2ї категорії

Наступну сторінку ми спроектували орієнтуючись на вже існуючу сторінку наших курсів. Проте замість виділення окремої сторінки на «Запис на курс» и їх об'єднали, тим самим зменшивши глибину вузлів для цільової дії (рис.4.11). Проте в іншому варіанті прототипів ми навпаки, розділили сторінки для того, щоб протестувати різні підходи до проектування інтерфейсу для користувачів «новачків».

Цільовий блок «Запис на курс» було спеціально відділено контрастним кольором, щоб звернути увагу та направити погляд користувача.

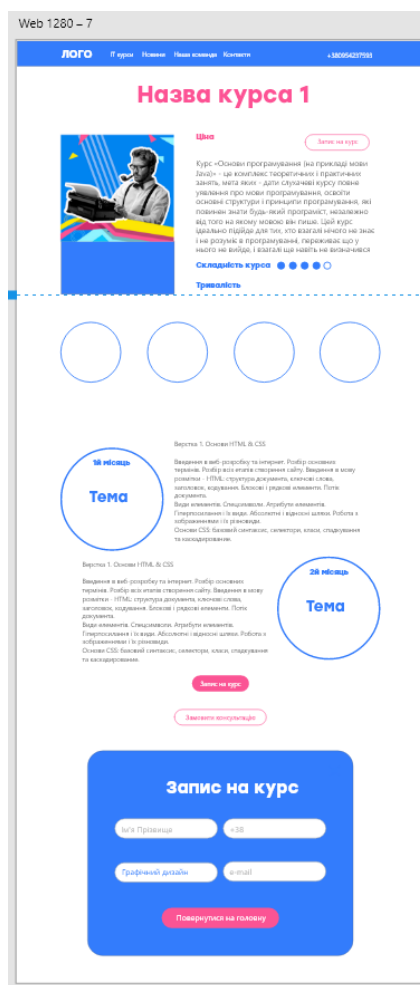


Рисунок 4.10- Реалізація зв'язків між сторінками сайту ІТ курсів. Варіант 2

Для того, щоб наблизити тестування до реальних умов ми використовували реальну загальну інформацію, і щоб користувач сформував думку про розміщення певних структурних блоків.

І після створення прототипів ми знову додали зв'язки між сторінками(рис 4.11).

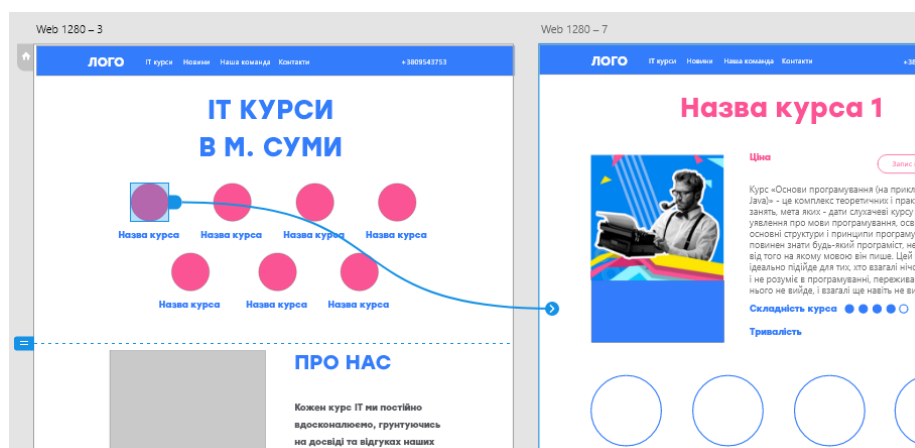


Рисунок 4.11- Створення переходів між сторінками

## 4.5 Реалізація рішення задачі оптимізації

На основі описаної вище математичної моделі, ми вносимо данні в MS Excel. А також потрібно розподілити всі формули відповідно до кількості вершин. У нашому випадку ми не будемо розглядати весь граф подій для сайту ІТ курсів(рис. 3.10), а візьмемо для прикладу лише його частину, яка містить кілька видів вершин та одразу їх пронумеруємо (рис. 4.12).

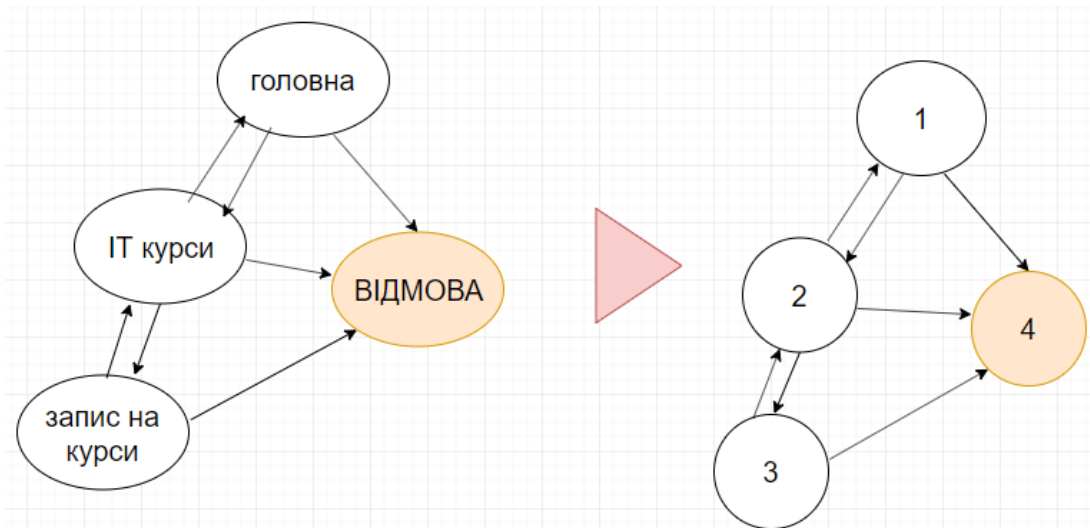


Рисунок 4.12- Частина графа подій

На першому етапі ми заносимо до файлу значення ймовірностей переходів, які були отримані під час тестування цільової аудиторії. Відповідно до кластеризованих груп ми виділяємо по 2 групи варіантів, які міститимуть по 2 альтернативні варіанти

		1 тип				2 тип					
I варіант(1)		Г	К	З	О	I варіант(2)		Г	К	З	О
	Г	0	0,2	0	0,8		Г	0	0,5	0	0,5
	К	0	0	0,8	0,2		К	0,1	0	0,4	0,5
	З	0	0,2	0	0,8		З	0	0,6	0	0,4
	О	0	0	0	1		О	0	0	0	1
II варіант(3)		Г	К	З	О	II варіант(4)		Г	К	З	О
	Г	0	0,1	0	0,75		Г	0	0,5	0	0,5
	К	0,3	0	0,2	0,5		К	0,3	0	0,6	0,1
	З	0	0	0	1		З	0	0,2	0	0,7
	О	0	0	0	1		О	0	0	0	1

Рисунок 4.13- Внесення даних до матриць переходів

Після цього створюємо змінні, за рахунок яких буде рахувати функція оптимізації, та одразу плануємо на них обмеження (рис. 4.14).

	$X_1^1$	$X_1^2$	$X_1^3$	$X_1^4$	$X_2^1$	$X_2^2$	$X_2^3$	$X_2^4$	$X_3^1$	$X_3^2$	$X_3^3$	$X_3^4$
	0	0,000	1,000	0,000	1,100	0,000	0,000	0,000	0,000	0,000	0,000	1,200
мин	0	0	0	0	0	0	0	0	0	0	0	0
	$\delta_1^1$	$\delta_1^2$	$\delta_1^3$	$\delta_1^4$	$\delta_2^1$	$\delta_2^2$	$\delta_2^3$	$\delta_2^4$	$\delta_3^1$	$\delta_3^2$	$\delta_3^3$	$\delta_3^4$
	0	0	1	0	1	0	0	0	0	0	0	1
маж	1	1	1	1	1	1	1	1	1	1	1	1
Значення	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле

Рисунок 4.14- Внесення даних до матриць переходів

Наступним кроком ми вносимо цільову функцію, яку будемо максимізувати симплекс-методом.

<b>Цільова функція</b>	$X_1^1 * P_{12}^1 + X_1^2 * P_{12}^2 + X_1^3 * P_{12}^3 + X_1^4 * P_{12}^4 + X_2^1 * P_{23}^1 + X_2^2 * P_{23}^2 + X_2^3 * P_{23}^3 + X_2^4 * P_{23}^4 \rightarrow \max$	<b>0,98</b>	Максимізуємо
------------------------	--	-------------	--------------

Рисунок 4.15- Внесення даних цільової функції

І в останню чергу ми прописуємо всі обмеження на кожен вершину (рис 4.16).

Обмеження для ЦФ							
			$X_1^1 + X_1^2 + X_1^3 + X_1^4 - (X_2^1 * P_{21}^1 + X_2^2 * P_{21}^2 + X_2^3 * P_{21}^3 + X_2^4 * P_{21}^4)$	1	=	1	
			$X_2^1 + X_2^2 + X_2^3 + X_2^4 - (X_1^1 * P_{12}^1 + X_1^2 * P_{12}^2 + X_1^3 * P_{12}^3 + X_1^4 * P_{12}^4)$	1	=	1	
			$X_3^1 + X_3^2 + X_3^3 + X_3^4 - (X_1^1 * P_{23}^1 + X_1^2 * P_{23}^2 + X_1^3 * P_{23}^3 + X_1^4 * P_{23}^4)$	1	=	1	
			$X_2^1 * P_{14}^1 + X_2^2 * P_{14}^2 + X_2^3 * P_{14}^3 + X_2^4 * P_{14}^4 + X_3^1 * P_{24}^1 + X_3^2 * P_{24}^2 + X_3^3 * P_{24}^3 + X_3^4 * P_{24}^4$	1	=	1	
$\delta_{11}$			$\delta_1^1 + \delta_1^2 + \delta_1^3 + \delta_1^4$	1	=	1	
			$\delta_2^1 + \delta_2^2 + \delta_2^3 + \delta_2^4$	1	=	1	
			$\delta_3^1 + \delta_3^2 + \delta_3^3 + \delta_3^4$	1	=	1	
			$x_1^{(k)} - W * \delta_1^{(k)} \leq 0,$ $x_1^{(k)} - w * \delta_1^{(k)} \geq 0:$				
	$W=$	10000	$0 \leq$	0	$-9998,9 \leq$	0	$0 \leq$
	$w=$	0,001	$0 \geq$	0	$1,099 \geq$	0	$0 \geq$
			$0 \leq$	0	$0 \leq$	0	$0 \leq$
			$0 \geq$	0	$0 \geq$	0	$0 \geq$
			$-9999 \leq$	0	$0 \leq$	0	$0 \leq$
			$0,999 \geq$	0	$0 \geq$	0	$0 \geq$
			$0 \leq$	0	$0 \leq$	0	$-9998,8 \leq$
			$0 \geq$	0	$0 \geq$	0	$1,199 \geq$

Рисунок 4.16- Формування обмежень

Після того як сформовано дані, ми за допомогою надбудови «Пошук рішень» вносимо обмеження у відповідності з комірок до системи (рис 4.18).

Обмеження для ЦФ		Обмеження для ЦФ			
$X_1^1 + X_1^2 + X_1^3 + X_1^4 - (X_2^1 * P_{21}^1 + X_2^2 * P_{21}^2 + X_2^3 * P_{21}^3 + X_2^4 * P_{21}^4)$		=B23+C23+D23+E23-(F23*C9+G23*K9+H23*C16+I23*K16)	=	1	
$X_2^1 + X_2^2 + X_2^3 + X_2^4 - (X_1^1 * P_{12}^1 + X_1^2 * P_{12}^2 + X_1^3 * P_{12}^3 + X_1^4 * P_{12}^4)$		=F23+G23+H23+I23-(B23*D8+C23*L8+D23*D15+E23*L15)	=	1	
$X_3^1 + X_3^2 + X_3^3 + X_3^4 - (X_1^1 * P_{23}^1 + X_1^2 * P_{23}^2 + X_1^3 * P_{23}^3 + X_1^4 * P_{23}^4)$		=J23+K23+L23+M23-(B23*E9+C23*M9+D23*E16+E23*M16)	=	1	
$X_2^1 * P_{14}^1 + X_2^2 * P_{14}^2 + X_2^3 * P_{14}^3 + X_2^4 * P_{14}^4 + X_3^1 * P_{24}^1 + X_3^2 * P_{24}^2 + X_3^3 * P_{24}^3 + X_3^4 * P_{24}^4$		=F23*F8+N8*G23+H23*F15+I23*N15+J23*F9+K23*N9+L23*F16+M23*N16	=	1	
$\delta_1^1 + \delta_1^2 + \delta_1^3 + \delta_1^4$		=J27+K27+L27+M27	=	1	
$\delta_2^1 + \delta_2^2 + \delta_2^3 + \delta_2^4$		=B27+C27+D27+E27	=	1	
$\delta_3^1 + \delta_3^2 + \delta_3^3 + \delta_3^4$		=F27+G27+H27+I27	=	1	
	$x_1^{(k)} - M * \delta_1^{(k)} \leq 0,$ $x_1^{(k)} - m * \delta_1^{(k)} \geq 0:$		$x_2^{(k)} - M * \delta_2^{(k)} \leq 0,$	$x_2^{(k)} - m * \delta_2^{(k)} \geq 0:$	$x_3^{(k)} - M * \delta_3^{(k)} \leq 0,$ $x_3^{(k)} - m * \delta_3^{(k)} \geq 0:$
<b>M=</b>	10000	=B23-C ≤ 0	=F23-C49 ≤ 0		=J23-C49*I ≤ 0
<b>m=</b>	0,001	=B23-C ≥ 0	=F23-C50 ≥ 0		=J23-C50*I ≥ 0
		=C23-C ≤ 0	=G23-C49 ≤ 0		=K23-C49*K ≤ 0
		=C23-C ≥ 0	=G23-C50 ≥ 0		=K23-C50*K ≥ 0
		=D23-C ≤ 0	=H23-C49 ≤ 0		=L23-C49*L ≤ 0
		=D23-C ≥ 0	=H23-C50 ≥ 0		=L23-C50*L ≥ 0
		=E23-C ≤ 0	=I23-C49 ≤ 0		=M23-C49*M ≤ 0
		=E23-C ≥ 0	=I23-C50 ≥ 0		=M23-C50*M ≥ 0

Рисунок 4.17- Формульний вигляд обмежень

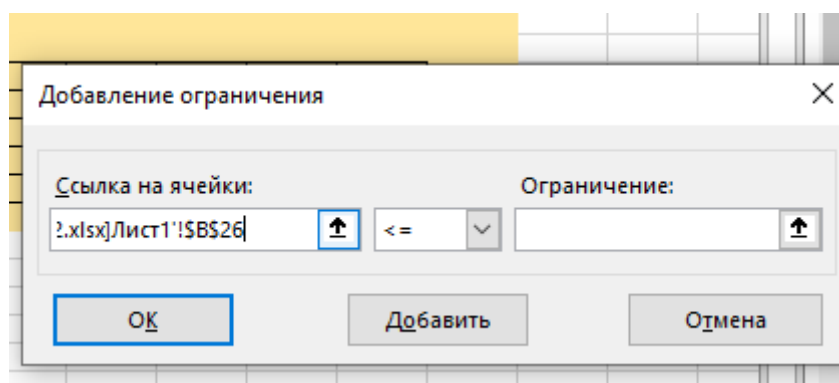


Рисунок 4.18- Внесення обмежень

І в результаті ми отримуємо більше 50 внесених в систему обмежень(рис. 4.19). Обираємо симплекс-метод, яким будемо розв'язувати задачу та визначаємо комірку, яку будемо максимізувати.

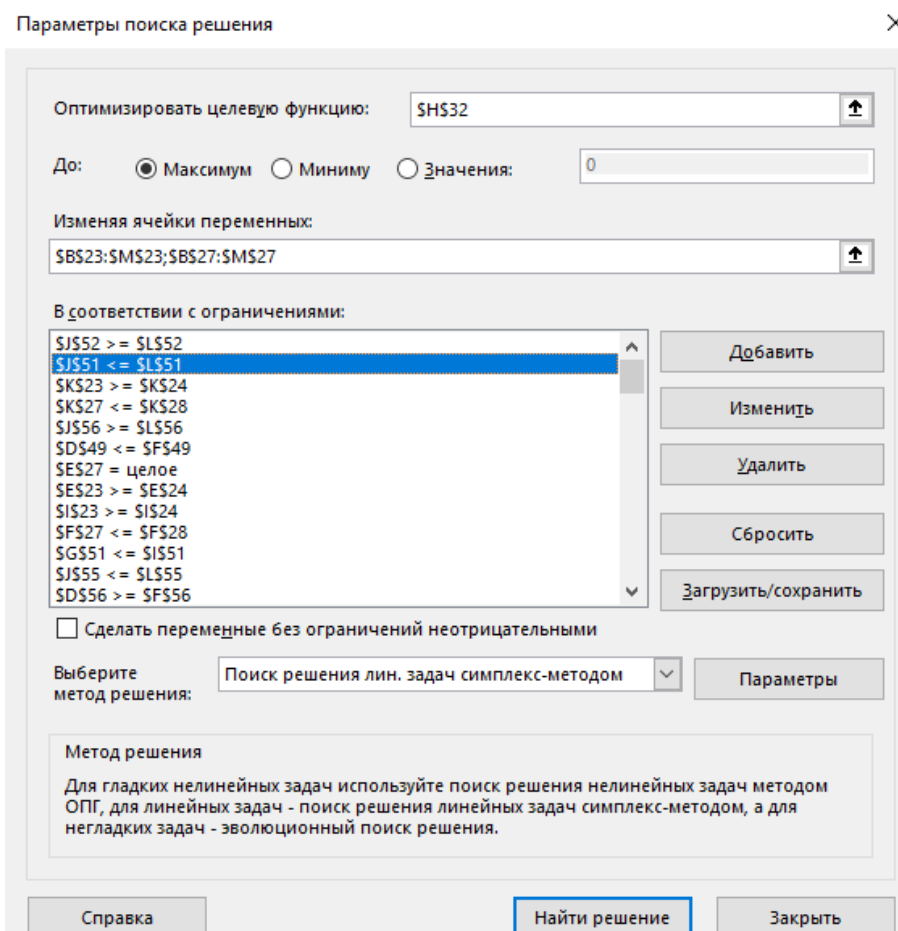


Рисунок 4.19- Вікно пошуку рішень

В результаті обчислень ми отримуємо значення цільової функції рівне 0.98.

Змінні нам показують що, оптимальним варіантом для вершини 1 2го типу є спосіб №1, а для вершин 1 типу також спосіб №1 (рис 4.20-4.21)

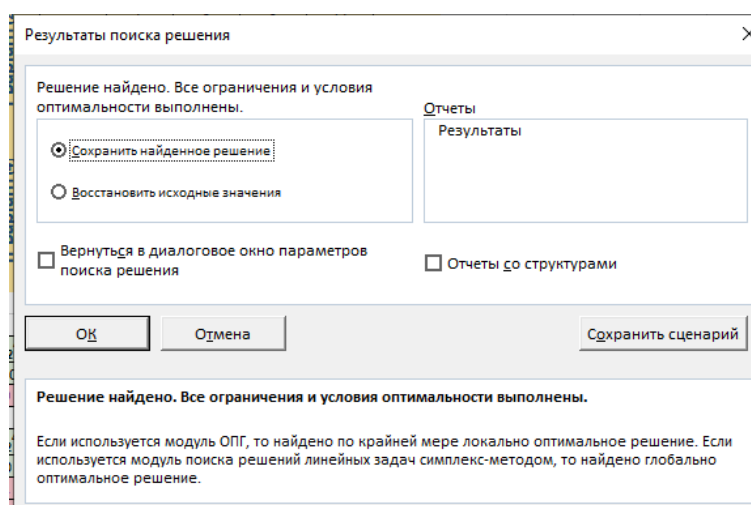


Рисунок 4.20- Вікно результату



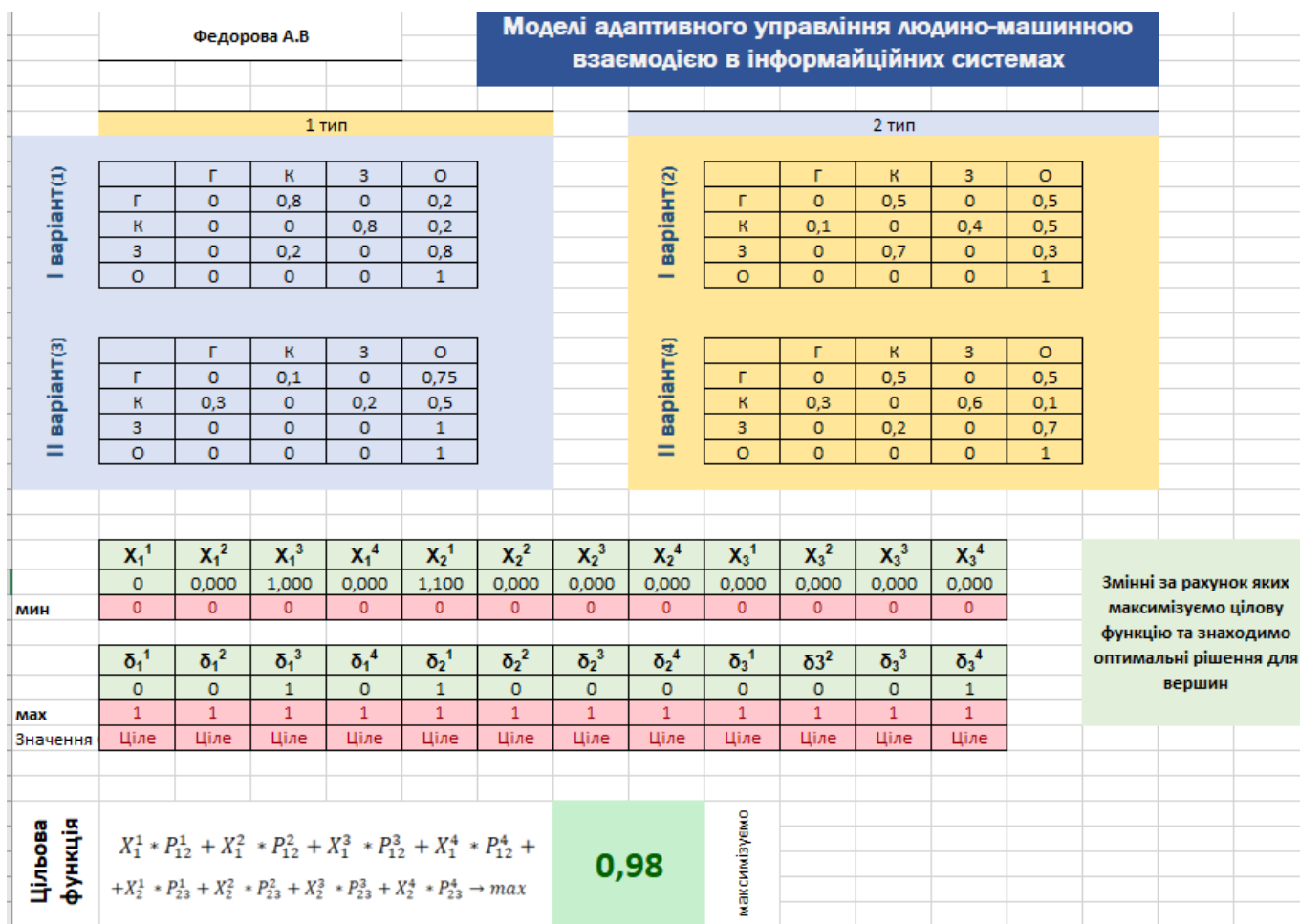


Рисунок 4.21- Результат пошуку цільової функції.

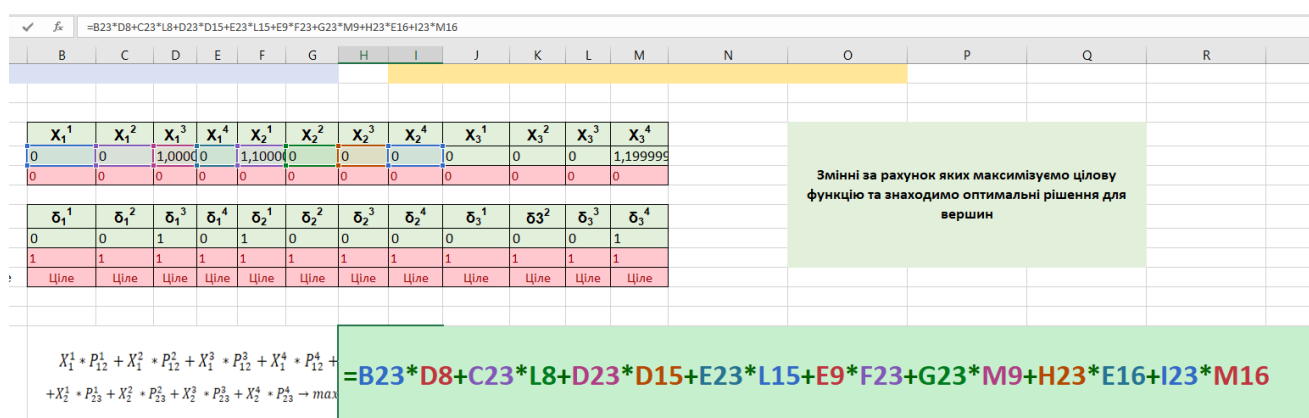


Рисунок 4.22- Формульний вигляд цільової функції.

Після цього для достовірності отриманих значень та оптимальних вершин ми зробимо ще кілька експериментів, змінюючи дані в матрицях

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
II варіант(3)			Г	К	З	О								
	Г	0	0,1	0	0,75									
	К	0,3	0	0,2	0,5									
	З	0	0	0	1									
	О	0	0	0	0	1								
II варіант(4)			Г	К	З	О								
	Г	0	0,5	0	0,5									
	К	0,9	0	0,6	0,1									
	З	0	0,2	0	0,7									
	О	0	0	0	0	1								
		$X_1^1$	$X_1^2$	$X_1^3$	$X_1^4$	$X_2^1$	$X_2^2$	$X_2^3$	$X_2^4$	$X_3^1$	$X_3^2$	$X_3^3$	$X_3^4$	
мин		0	0,000	1,000	0,000	1,100	0,000	0,000	0,000	0,000	0,000	0,000	1,200	
		0	0	0	0	0	0	0	0	0	0	0	0	
		$\delta_1^1$	$\delta_1^2$	$\delta_1^3$	$\delta_1^4$	$\delta_2^1$	$\delta_2^2$	$\delta_2^3$	$\delta_2^4$	$\delta_3^1$	$\delta_3^2$	$\delta_3^3$	$\delta_3^4$	
мах		0	0	1	0	1	0	0	0	0	0	0	1	
		1	1	1	1	1	1	1	1	1	1	1	1	
Значення		Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	
Цільова функція	$X_1^1 * P_{12}^1 + X_1^2 * P_{12}^2 + X_1^3 * P_{12}^3 + X_1^4 * P_{12}^4 +$ $+ X_2^1 * P_{23}^1 + X_2^2 * P_{23}^2 + X_2^3 * P_{23}^3 + X_2^4 * P_{23}^4 \rightarrow \max$							0,78		максимізуємо				

Рисунок 4.23- Результат експерименту 1

	1 тип								2 тип					
I варіант(1)			Г	К	З	О								
	Г	0,1	0,3	0	0,6									
	К	0	0	0,8	0,2									
	З	0	0,2	0	0,8									
	О	0	0	0	0	1								
II варіант(3)			Г	К	З	О								
	Г	0	0,1	0	0,75									
	К	0,3	0	0,2	0,5									
	З	0	0	0	1									
	О	0	0	0	0	1								
I варіант(2)			Г	К	З	О								
	Г	0	0,5	0	0,5									
	К	0,5	0	0,1	0,1									
	З	0	0,6	0	0,4									
	О	0	0	0	0	1								
II варіант(4)			Г	К	З	О								
	Г	0	0,6	0	0,4									
	К	0,9	0	0,6	0,1									
	З	0	0,3	0	0,7									
	О	0	0	0	0	1								
		$X_1^1$	$X_1^2$	$X_1^3$	$X_1^4$	$X_2^1$	$X_2^2$	$X_2^3$	$X_2^4$	$X_3^1$	$X_3^2$	$X_3^3$	$X_3^4$	
мин		0	0,000	1,000	0,000	1,100	1,000	0,000	0,000	0,000	0,000	0,000	0,000	
		0	0	0	0	0	0	0	0	0	0	0	0	
		$\delta_1^1$	$\delta_1^2$	$\delta_1^3$	$\delta_1^4$	$\delta_2^1$	$\delta_2^2$	$\delta_2^3$	$\delta_2^4$	$\delta_3^1$	$\delta_3^2$	$\delta_3^3$	$\delta_3^4$	
мах		0	0	1	0	1	0	0	0	0	0	0	1	
		1	1	1	1	1	1	1	1	1	1	1	1	
Значення		Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	Ціле	
Цільова функція	$X_1^1 * P_{12}^1 + X_1^2 * P_{12}^2 + X_1^3 * P_{12}^3 + X_1^4 * P_{12}^4 +$ $+ X_2^1 * P_{23}^1 + X_2^2 * P_{23}^2 + X_2^3 * P_{23}^3 + X_2^4 * P_{23}^4 \rightarrow \max$								0,86		максимізуємо			

Рисунок 4.24- Результат експерименту 2

З цільової функції помітно, що найбільшого значення вона досягає з варіанту 1

Після проведеної оптимізації було вирішено повторно протестувати інтерактивні прототипи сайту ІТ курсів ти ми знову зробили кластеризацію отриманих даних.

Таблиця 4.2-Результати отриманих даних від користувачів повторного тестування

$T_0$	$K_0$	$T_0$	$K_0$
10	9.3	30	12.5
20	11.8	62	12.6
14	12.7	65	9.2
18	10.7	68	11.3
13	8.2	42	13.4
22	10.4	15	11.1
60	7.2	20	12.3
11	12.4	44	10.2
15	10.4	75	11.8
17	9.9	80	7.3

Ми отримали доволі успішний розподіл, користувачі оцінили оновлені інтерфейси на вищі бали ніж минулого разу. Помітно, що на графіку присутні викиди даних, проте несуттєві, не дивлячись на час проведений на сайті, користувачі все одно оцінили його краще по 4м факторам ніж на попередньому тестуванні(рис. 4.24).

Ми кластеризацію користувачів робили в середовищі MatLab.

Код програми наведено в додатку Б

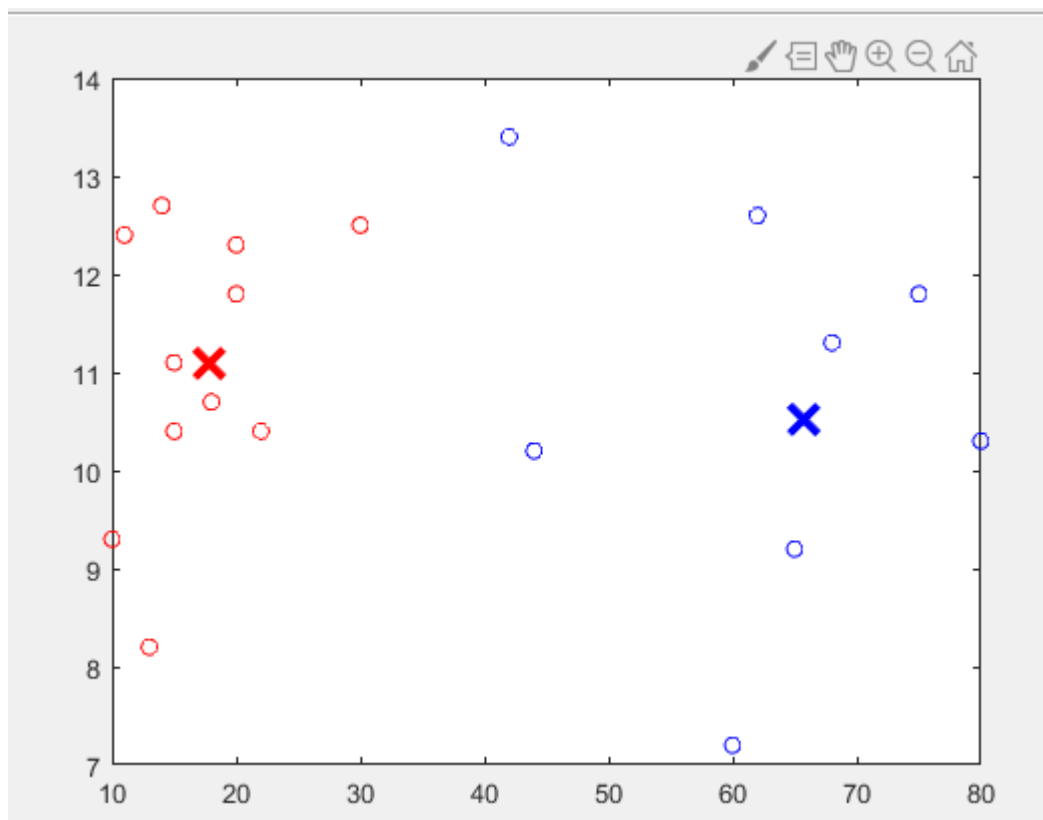


Рисунок 4.25- Результат повторної кластеризації користувачів.

#### 4.6 Аналіз обраних прототипів

Для 1ї категорії користувачів(більш досвідченої) ми обрали варіант структури зображений на рис 4.26-4.27. Тут у користувача одразу є можливість переглянути всі доступні курси від компанії, відсутня зайва інформація.

На цій сторінці одразу є найважливіша інформація щодо курсів(назва, ціна, старт) та є можливість запису по натисканню на кнопку «Запис на курс»

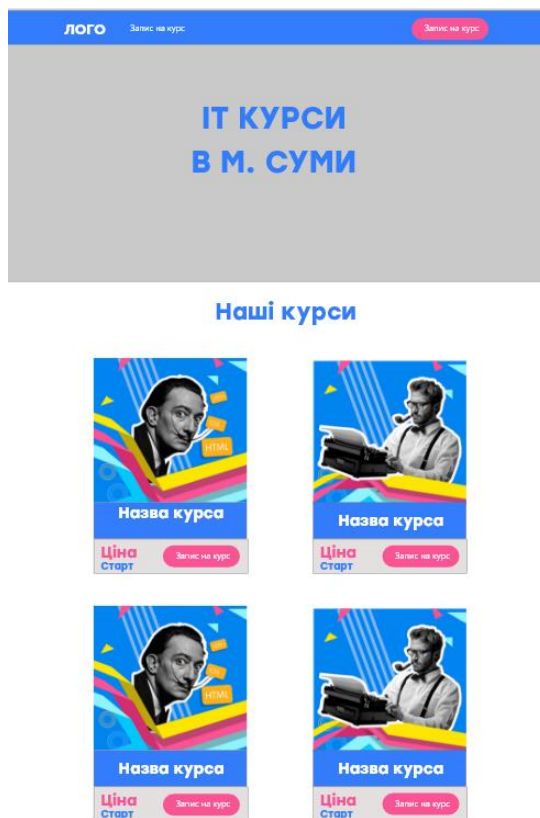


Рисунок 4.26- Приклад прототипу «Головної» сторінка

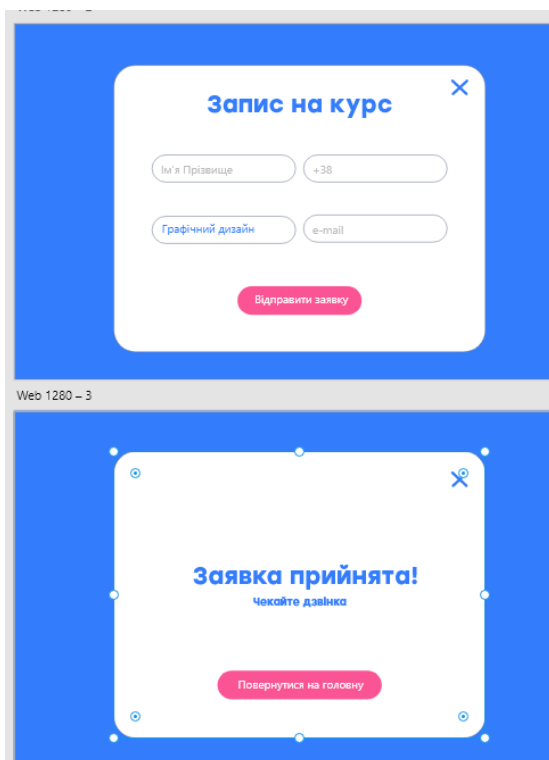


Рисунок 4.27- Приклад прототипу «Запис на курс» сторінка

Для 2ї категорії користувачів ми обрали варіант рис 4.26-4.27

У цьому випадку ми створили перший екран головної сторінки, який містить перехід на сторінку «Курсів».

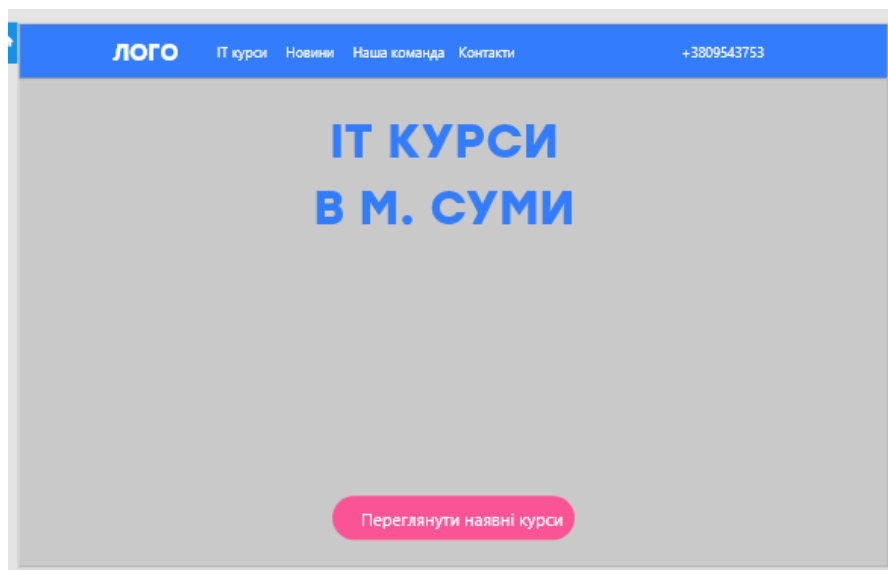


Рисунок 4.26- Приклад прототипу «Головна» сторінка

Для сторінки «Курсів» ми виділили дві окремі сторінки. Перша з переліком всіх актуальних курсів. Друга сторінка з детальним описом конкретного курсу, де присутня кнопка «Запис на курс». Також на сторінку для категорії «новачків» ми додали кнопку «Замовити консультацію». Деякі користувачі відмітили зручність такої кнопки, оскільки для недосвідчених користувачів, які хочуть швидко отримати відповіді на свої запитання, консультація від робітників буде дуже актуальною. Більш детально переглянути прототипи можна в додатку В.

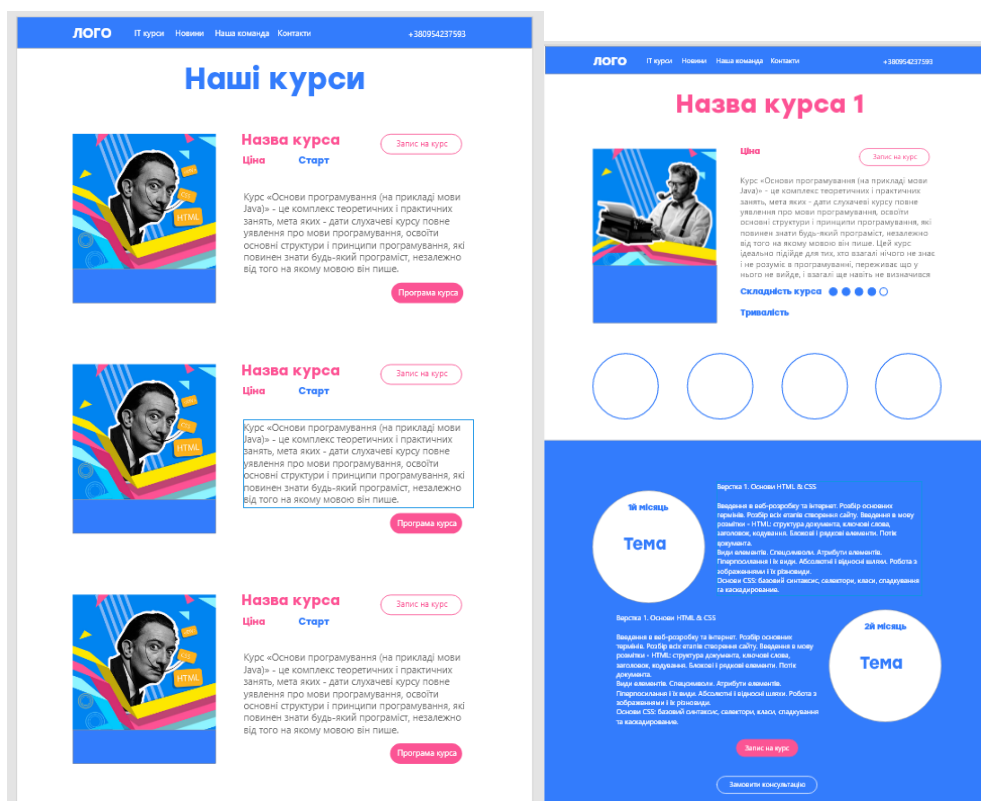


Рисунок 4.27- Приклад прототипу сторінки «Курси»

Для сторінки «Запису на курс» ми вирішили використовувати впливаюче вікно з формою запису, яке має два стани (рис. 4.28).

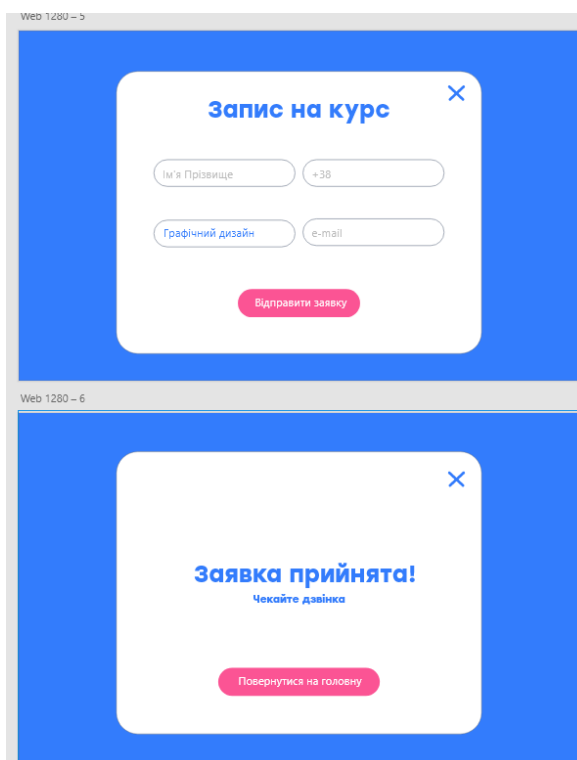


Рисунок 4.28- Приклад прототипу сторінки «Курси»

## ВИСНОВОК

Оцінка юзабіліті на сьогоднішній день є актуальною проблемою оцінки інформаційних систем в цілому.

Для оцінки цього показника існує багато різних за складністю методів. Одним з таких методів оцінки можна назвати процеси Маркова, які дозволяють чітко змодельовати роботу будь якої системи. За допомогою цих моделей можна провести кількісне порівняння зручності користування, навіть якщо проект ще тільки починає розроблюватися.

Але, майже всі методи можуть лише статично оцінити вашу систему та не враховують необхідності в адаптації.

Існуючі методи до побудови механізму адаптації в інформаційних системах не можуть забезпечити можливість реалізації адаптивності в системі.

Дослідження розвинуте в магістерській роботі, дозволило описати матриці ймовірностей для станів та переходів, які потім були використані для оцінки оптимальних варіантів інтерфейсу для різних груп користувачів. Розподіл цільової аудиторії ми здійснювали за допомогою факторного аналізу, який також дозволяє виявляти найслабші місця системи відповідно до 4х факторів.

Отримана математична модель дозволяє оцінити складність інтерфейсу, контроль станів помилок та ефективність.

Задача адаптації була зведена до пошуку розв'язку оптимізаційної функції, яка описує максимуми ймовірностей переходу по системі для заданих груп користувачів.

Проведені дослідження повністю підтвердили раціональність такого підходу.



## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Fuzzy c-means кластеризація. Кіберфорум: веб-сайт. URL: <http://www.cyberforum.ru/matlab/thread1736398.html>
2. Метод нечіткої кластеризації С-середніх. Вікіпедія: веб-сайт. URL: <https://ru.wikipedia.org/wiki>
3. Берестецька О. Figma: огляд програми: веб-сайт. URL: <https://sendpulse.com/ru/blog/figma>
4. Марковські процеси. Бібліотека: веб-сайт. URL: [http://e-biblio.ru/book/bib/06\\_management/teor\\_mass\\_obslug/158.9.00.html](http://e-biblio.ru/book/bib/06_management/teor_mass_obslug/158.9.00.html)
5. Шерстенников Ю.В. Застосування моделей марківського процесу до досліджень економічної ефективності роботи фірми: тези. Дніпропетровськ, 2007. 95с. URL: <http://dspace.nbuv.gov.ua/bitstream/handle/123456789/15957/14-Sherstnikov.pdf?sequence=1>
6. Горбунов А.Л. Марківські процеси для відвідування сайту: тези. Москва, 2011. С. 68-73 URL: [http://elar.urfu.ru/bitstream/10995/1334/1/IMAT\\_2007\\_08.pdf](http://elar.urfu.ru/bitstream/10995/1334/1/IMAT_2007_08.pdf)
7. Бакаєв М.А. Сучасні тенденції в автоматизованій оцінці юзабіліті і факторів поведінки в алгоритмах пошукових систем: стаття, Новосибірськ, 2017.С.447-454 URL:[https://www.researchgate.net/publication/319466478\\_Sovremennye\\_tendencii\\_v\\_avtomatizirovannoј\\_ocenke\\_uzabiliti\\_i\\_povedenceskie\\_factory\\_v\\_algoritmah\\_poiskovyh\\_sistem](https://www.researchgate.net/publication/319466478_Sovremennye_tendencii_v_avtomatizirovannoј_ocenke_uzabiliti_i_povedenceskie_factory_v_algoritmah_poiskovyh_sistem)
8. Чернов В.П., Іванівський В.Б. Теорія масового обслуговування . Москва: ИНФРА-М, 1998. 158 с.
9. 11. Куравський Л. С., Баранов С. Н. Дискримінантні мережі в задачах діагностики. - Нейрокомп'ютери: розробка і застосування, 2003 №8-9, с. 3-9.
10. Вентцель Е.С., Овчаров Л.А. Дослідження операцій. Завдання, принципи, методологія. Москва: Вища. школа, 2001. 208 с.
11. Вентцель Е.С. Теорія випадкових процесів і її інженерні додатки. Москва: Вища. школа; 2007. 479 с.

12. 7. Гмурман В.С. Теория вероятностей и математическая статистика Москва.: Высшая школа, 1998. 479 с.
13. Кориков А. М. Система «Людина - комп'ютер»: на шляху створення людино-орієнтованого інтерфейсу. Томськ: В-Спектр, 2010. 183 с.
14. Jansen B., Sprink A., Saraevic T. Real Life, Real users and Real needs: a Study and Analysis of User Queries on the Web. Information Processing and Management. 36: 2000 Elsevier. P. 207-227.
15. 4. Баканов А. С. Ергономіка призначеного для користувача інтерфейсу: від проектування до моделювання людино-комп'ютерного взаємодії. Інститут психології РАН, 2011. 176 с.
16. Сергєєв С. Ф. Інженерна психологія і ергономіка: навчальний посібник. М .: НДІ шкільних технологій, 2008. 176 с.
17. Сергєєв С. Ф., Введення в інженерну психологію і ергономіку імерсивні середовища: навчальний посібник. СПб: ІТМО, 2011. 258 с.
18. Сергєєв С. Ф. Падерно П. І. Введення в проектування інтелектуальних інтерфейсів: навчальний посібник СПб: ІТМО, 2011. 108 с.
19. Сергєєв С. Ф. Юзабіліті-тестування інтерфейсів інформаційних систем в гуманітарних науках і мистецтві: навчальний посібник, 2012. 86 с.
20. Сергєєв С. Ф., Соколов В. Н., Коротєєв Г. Л. Методи юзабіліті в тренажерах і навчальних системах. 2012. 132 с.
21. Унгер Р., Чендлер К. UX-дизайн. Практичний посібник з проектування досвіду взаємодії. СПб: Символ-Плюс, 2011. 336 с.
22. Шлаєн П. Я., Львів В. М .. Ергономіка для інженерів. Ергономічне забезпечення проектування людино-машинних комплексів: проблеми, методологія, технології Твер: ТвГУ, 2008. 476 с.
23. Баканова Н. Б. Розробка інтерфейсів користувача корпоративних інформаційних систем Твер: Тріада, 2008. 112 с.

24. Kiravsky L.S. and Baranov S.A. The concept of multifactor Markov networks and its application to forecasting and diagnostics of technical systems. Proc. Condition Monitoring 2005, Cambridge, United Kingdom, pp. 111-117, July 2005.
25. Зінченко Т. П. Впізнання і кодування Львів: Вид-во ЛДУ, 1981. 183 с.
26. Калиновський А. І. Юзабіліті: як зробити сайт зручним. Мінськ: Нове знання, 2005. 220 с
27. Кірсанов Д. Веб-дизайн: книга Дмитра Кірсанова. СПб: Символ-Плюс, 2001. 376 с.
28. Купер А., Рейман Р., Кронін Д. Алан Купер про інтерфейс. Основи проектування взаємодії. СПб: СимволПлюс, 2009. 688 с.
29. Львів В. М, Шлаєн П. Я. Ергономіка. Вступний курс. Навчальний посібник для вузів. Твер: Тріада, 2004. 188 с.
30. Львів В. М, Магазанник В. Д. Людино-комп'ютерна взаємодія: навчальний посібник для вузів Твер: Тріада, 2005. 200 с
31. Нільсен Я. Веб-дизайн. Книга Якоба Нільсена СПб: Символ-Плюс, 2006. 512 с.
32. Нільсен Я., Лоранжер Х. Web-дизайн: зручність використання. Москва: Вільямс, 2007. 366 с.
33. Viterbi A.J. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory, vol.13, No.2, pp.260-269, 1967
34. Нільсен Я., Перніче К. Веб-дизайн: аналіз зручності використання веб-сайтів по руху очей. Москва: Вільямс, 2010. 480 с.
35. Норман Д. Дизайн звичних речей Москва: Вільямс, 2006. 384 с.
36. Раскін Д. Інтерфейс: нові напрямки в проектуванні комп'ютерних систем. СПб: Символ-Плюс, 2007. 272 с.
37. Kiravsky L.S. and Baranov S.A. Synthesis of Markov networks for forecasting fatigue failures. Proc. Condition Monitoring 2003, Oxford, United Kingdom, pp. 76-91, July 2003.

## ДОДАТОК А.

### А.1 Ідентифікація мети виконання проекту методом SMART

Узагальнена мета проекту полягає у розробці математичних моделей для вибору раціональних варіантів управління взаємодії людини та інформаційного сервіса .

Мета проекту: Розробити математичні моделі для варіантів реалізації адаптивності під групи користувачів. Показники будуть оцінюватися керівником та експертами з об'єкта дослідження. Дані математичні моделі дозволять отримати дані, за якими можна буде необхідні варіанти адаптації інформаційних сервісів.

Таблиця А.1 – Деталізація мети методом SMART

Specific (конкретна)	Розробити математичні моделі для адаптації людино-машинних систем
Measurable (вимірювана)	Знайти оптимальне рішення для побудови математичної моделі
Achievable (досяжна)	Зробити інформаційні системи доступнішими для різних груп аудиторії
Relevant (реалістична)	Створити модель яка промоделює різні варіанти роботи сайту відносно груп
Time-framed (обмежена у часі)	Створити математичні моделі до завершення термінів передзахисту

### А.2 Планування змісту структури робіт ІТ-проекту. WBS діаграма

На верхньому рівні WBS ми фіксуємо продукт проекту у моєму випадку це « Моделі адаптивного управління людино-машинною взаємодією в інформаційних системах». На першому рівні ми виділяємо 4 пункти робіт. На наступному II рівні ми виділяємо 4 основні пункти, які є основними етапами роботи над проектом. На

рис. А.1)показана описана модель. Всі 4 рівні моделі містять свої підпункти.

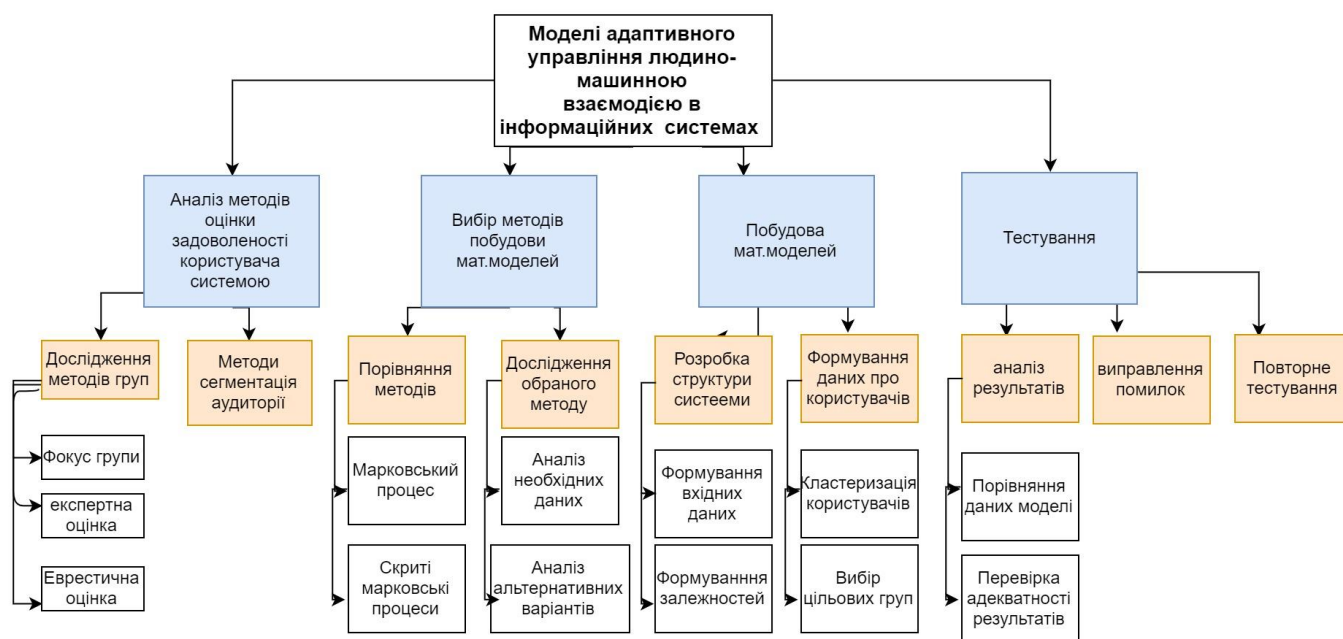


Рисунок А.1 – Діаграма WBS для запланованого проекту

### А.3 Побудова календарного графіку для планування термінів проекту

Для того щоб успішно реалізувати проект та запланувати час потрібний для реалізації кожного окремого етапу, ми опишемо тривалість виконання робіт у вигляді діаграми Ганта (рис. А.2).

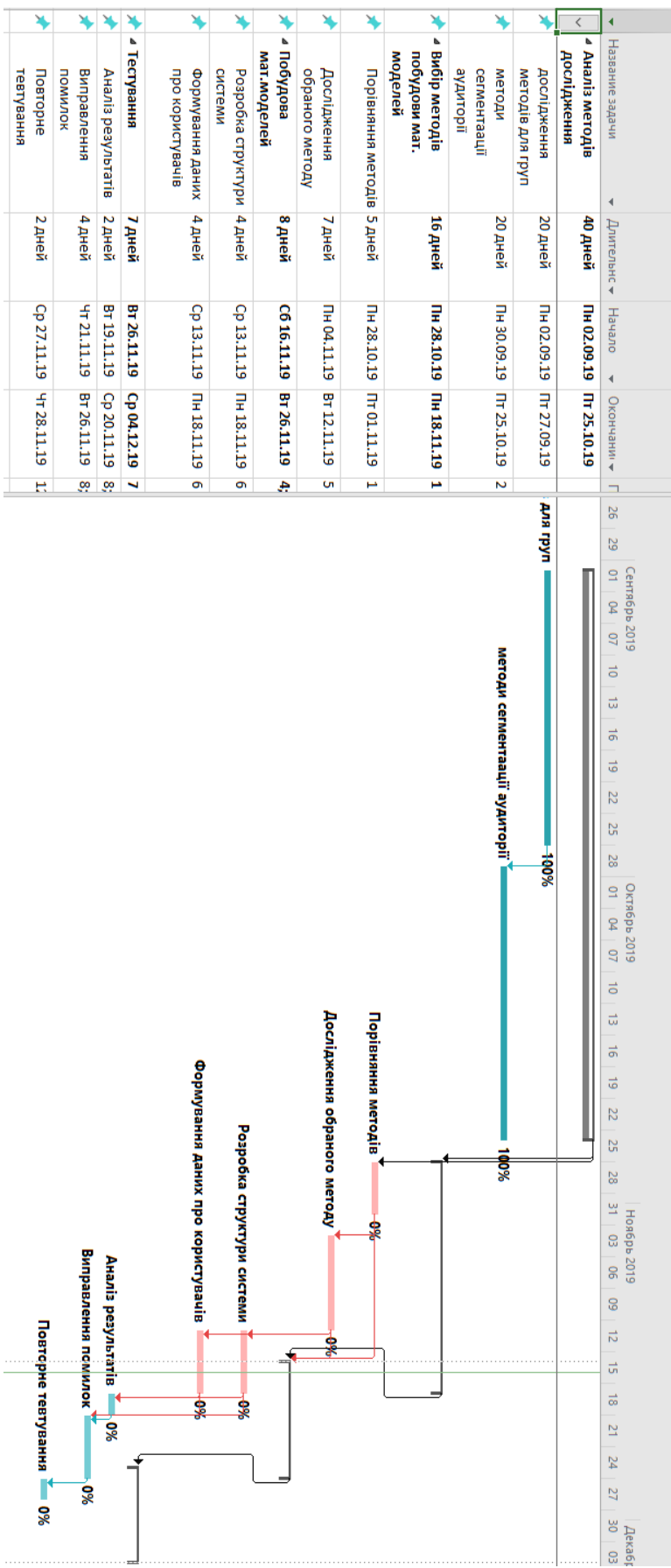


Рисунок А.2 – Фрагмент діаграми Ганта

## А.4 Планування ризиків проекту

Під час виконання проекту ми передбачаємо такі ризики (рис. А.3)

1	Зменшення дедлайнів проекту
2	Зміна цілей математичних моделей
3	Зміна вимог до отриманих результатів
4	Заміна однієї з моделей
5	Виникнення помилок при аналізі математичних моделей
6	Виникнення помилок при побудові математичних моделей
7	Виявлення помилок при аналізі результатів мат. Моделей
8	Отримання неприйнятних результатів від мат. моделей.

Рисунок А.3 – Виділені ризики

Для вирішення проблеми у разі виникнення запланованого ризику було передбачено відповідні плани дій (табл. А.2). Основними проблемами у випадку виникнення ризиків можуть бути зміщені терміни реалізації, відмова від додаткових функцій та збільшення бюджету проекту

Таблиця А.2- План дій при ризиках

Ризики	План А	План Б
Зменшення дедлайнів проекту	Відмова від додаткових функцій	Відмова від не основних задач
Зміна цілей математичних моделей	Збір нових даних	На пізньому етапі залишити без змін
Зміна вимог до отримуваних результатів	Дослідити лінії впливу, при впливі на одну з математичних моделей додати додаткові змінні	
Заміна однієї з моделей	Збір даних для нової моделі	Максимальна орієнтованість на заміну всіх даних, або на часткову заміну «специфічних даних»
Виникнення помилок аналізі математичних моделей	Виправлення помилок	
Виникнення помилок при побудові математичних моделей	Виправлення всіх помилок та перерахунок даних.	Виправлення лише критичних помилок та перерахунок даних
Аналізі результатів мат. моделей	Виявлення джерела помилок та виправлення	Виправлення тільки критичних помилок
Отримання неприйнятних результатів від обчислення ФМ	Початок пошуку помилок з введення алгоритма та виправлення на всіх подальших етапах	Якщо мало часу, аналіз джерела помилки, швидке виправлення помилки, або заміна не важливої функції

Після того як ми розробили плани дій, було побудовано таблицю ймовірностей виникнення ризику, а також таблицю втрат, які можуть бути спричинені ризиками(рис. А.4).

	виникнення	R1	R2	R3	R4	R5	R6	R7	R8
<40%	малоймовірні	+		+			+		+
40-60%	ймовірні		+					+	
>60%	можливі				+	+			

	втрати	R1	R2	R3	R4	R5	R6	R7	R8
до3 днів	мінімальні		+		+				
3- 7 днів	середні			+		+	+		
>7днів	максимальні	+						+	+

Рисунок А.4 – Таблиці з MS Excel

На рис А.5 можна побачити матрицю ймовірностей

мож	R4	R5				
ймов	R2		R7			
мал		R3,R6	R1,R8			
	мін	сер	макс			

	-	не важливі
	-	помірні
	-	критичні

Рисунок А.5 – Матриця ймовірності втрат

### А.5 Побудова матриці відповідальності

Матриця відповідальності будується для того, щоб визначити відповідальних осіб за кожен етап виконання проекту. . Створюємо її на основі схеми WBS (рис. А.6)

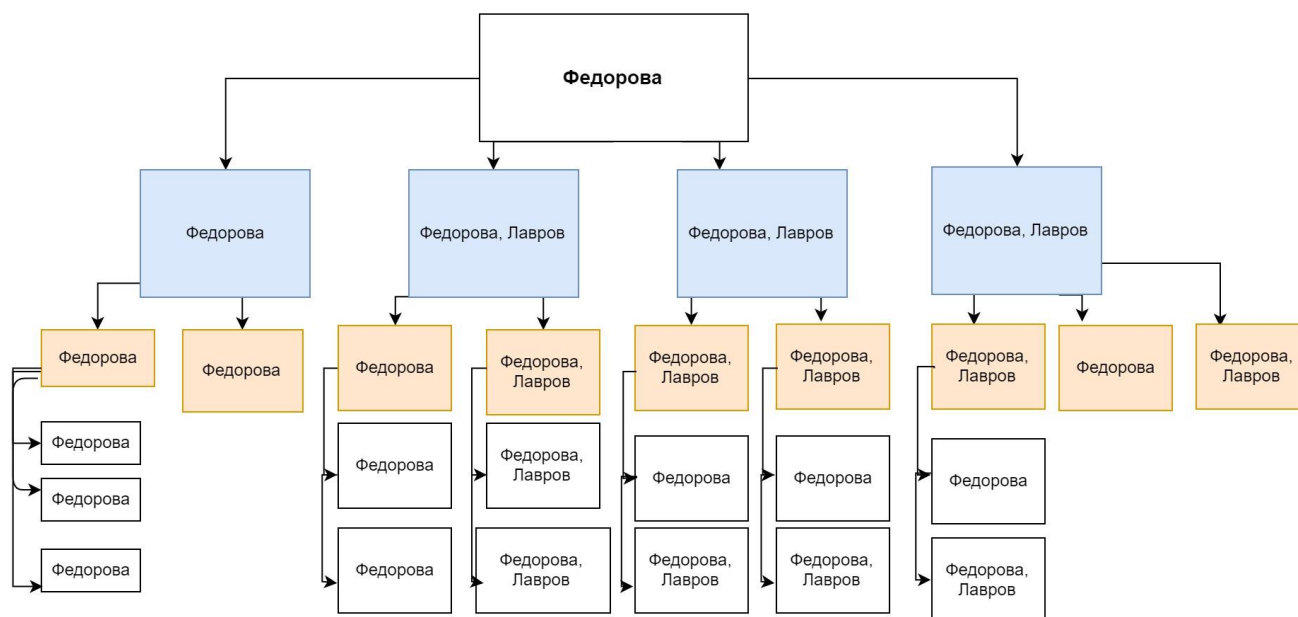


Рисунок А.6 – Діаграма OBS



## ДОДАТОК Б. РОЗВ'ЯЗАННЯ В МАТЛАВ

```
x_var = [10 8.1; 21 6.8; 15 5.7; 18 6.5; 13 8.2; 22 5.2; 63 7.2; 5 10.2; 30 7.5; 62
5.2; 70 6.2; 68 6.1; 75 5.4; 15 8.1; 20 8.3; 60 5; 85 4.8; 90 4.6]
```

```
[centers,U] = fcm(x_var,2); % делаем разбиение на 2 класса
```

```
maxU = max(U);
index1 = find(U(1,:) == maxU);
index2 = find(U(2,:) == maxU);
```

```
plot(x_var(index1,1),x_var(index1,2),'ob') % рисуем ваши 2 класса
hold on
plot(x_var(index2,1),x_var(index2,2),'or')
plot(centers(1,1),centers(1,2),'xb','MarkerSize',15,'LineWidth',3)
plot(centers(2,1),centers(2,2),'xr','MarkerSize',15,'LineWidth',3)
hold off
```

### Повторне тесування

```
x_var = [10 8.1; 20 11.8; 14 12.7; 18 8.5; 13 8.2; 22 10.4; 60 7.2; 5 10.4; 15 9.4;
30 12.5; 62 12.6; 65 9.2; 68 11.3; 42 13.4; 15 11.1; 20 12.3; 44 10.2; 75 8.8; 80
7.3]
```

```
[centers,U] = fcm(x_var,2); % делаем разбиение на 2 класса
```

```
maxU = max(U);
index1 = find(U(1,:) == maxU);
index2 = find(U(2,:) == maxU);
```

```
plot(x_var(index1,1),x_var(index1,2),'ob') % рисуем ваши 2 класса
hold on
plot(x_var(index2,1),x_var(index2,2),'or')
plot(centers(1,1),centers(1,2),'xb','MarkerSize',15,'LineWidth',3)
plot(centers(2,1),centers(2,2),'xr','MarkerSize',15,'LineWidth',3)
hold off
```

## ДОДАТОК В. ПРОТОТИПИ

### 1 категорія користувачів. Варіант 1

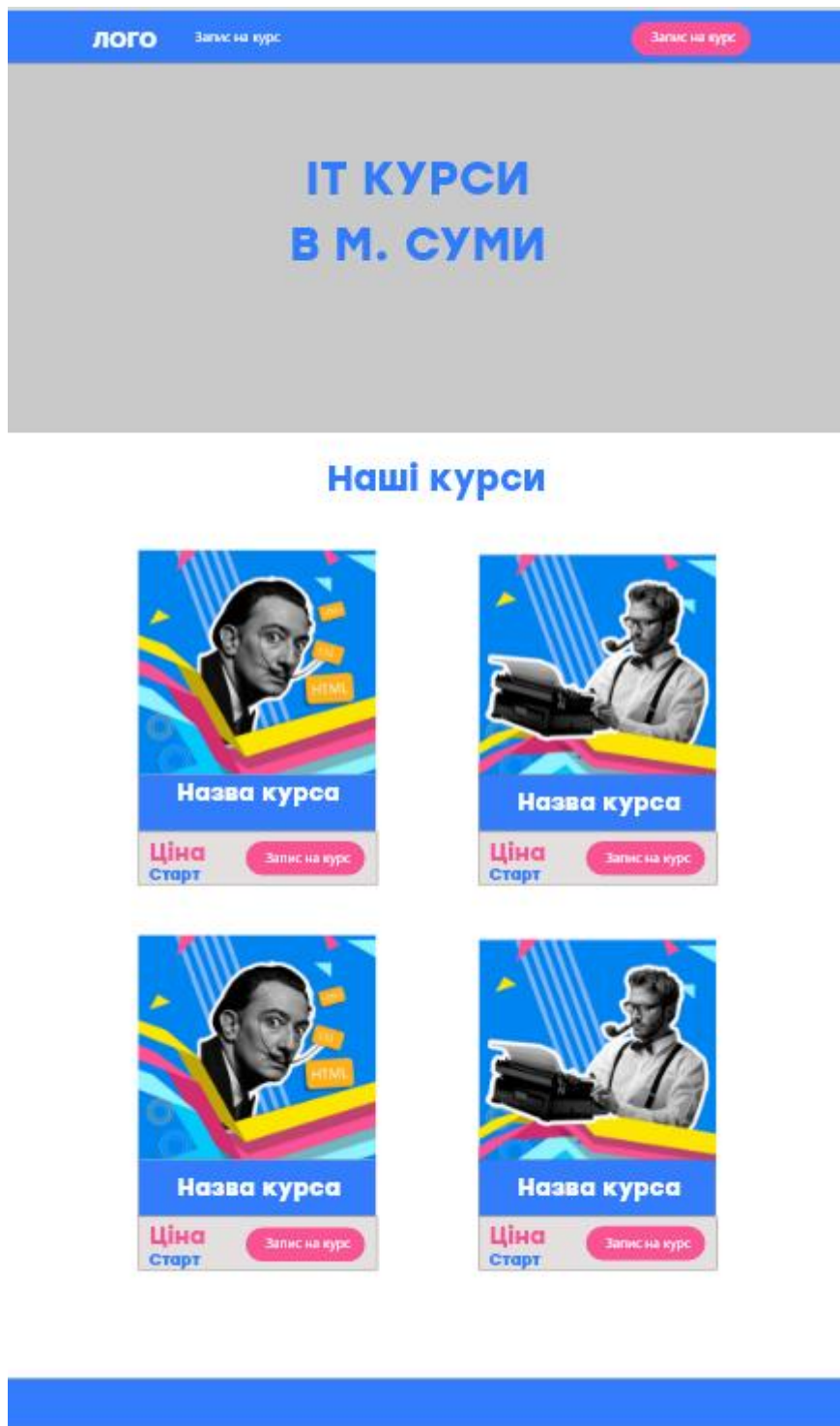


Рисунок Б.1 – Прототип «Головної» та сторінки «Курси»

Web 1280 – 2

## Запис на курс

Ім'я Прізвище

+38

Графічний дизайн

e-mail

[Відправити заявку](#)

Web 1280 – 3

## Заявка прийнята!

Чекайте дзвінка

[Повернутися на головну](#)

Рисунок Б.2– Прототип сторінки «Запис на курс»

## 1 категорія користувачів. Варіант 2

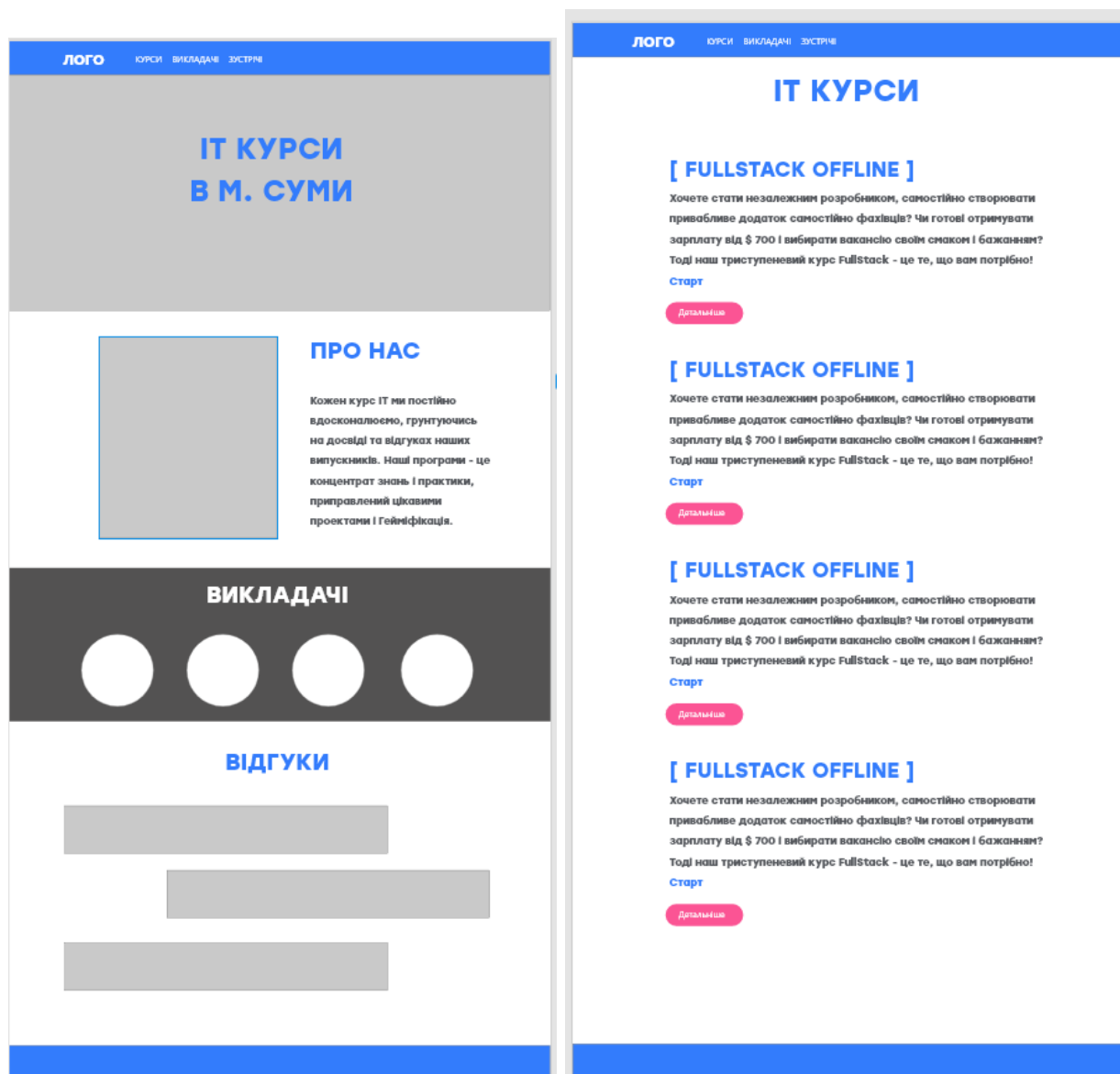


Рисунок Б.3 – Прототип «Головної» сторінки та сторінки «Курси»

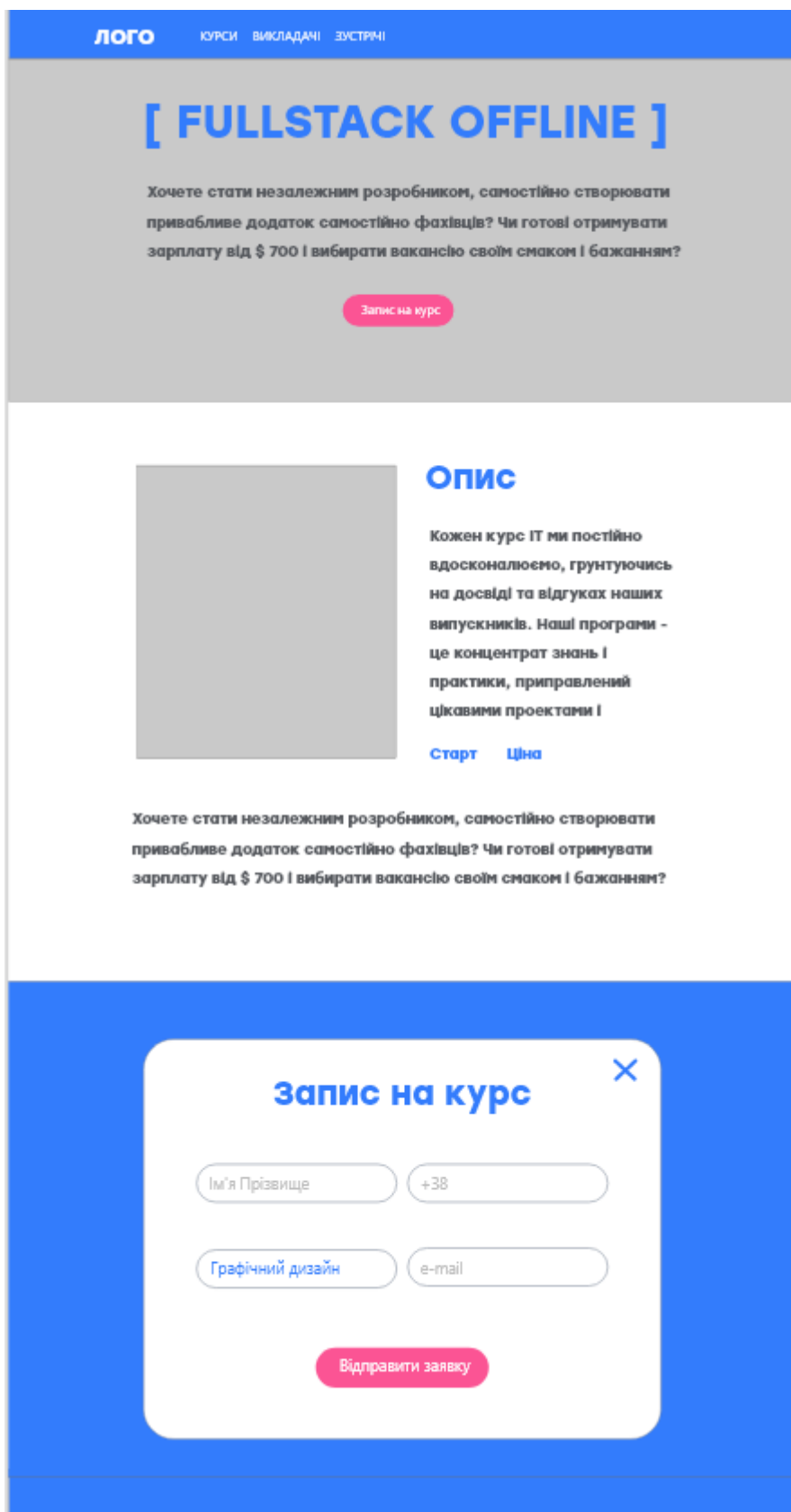


Рисунок Б.4 – Прототип сторінки «Курси в об'єднанні з запис на курс»

## 2категорія користувачів. Варіант 1

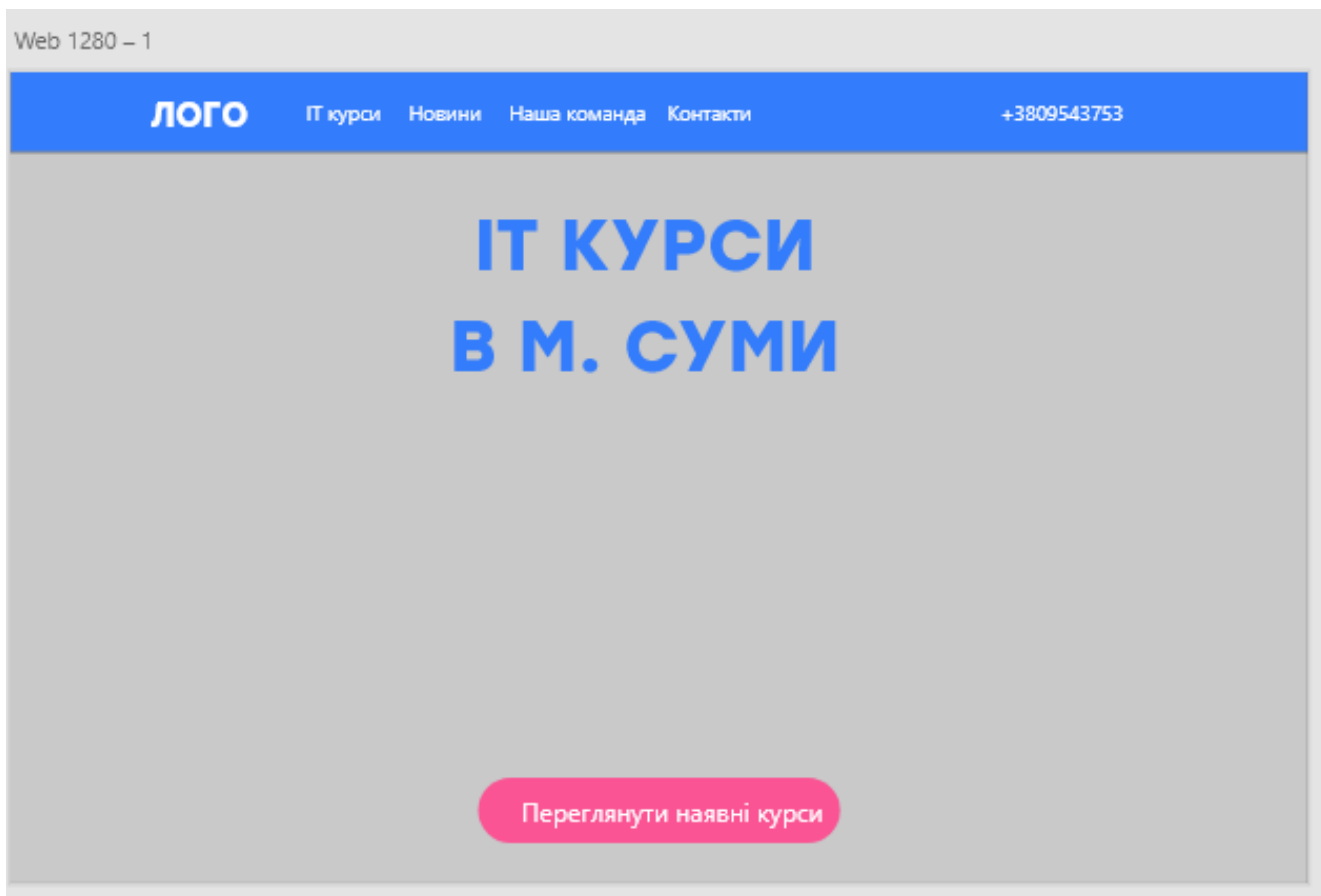


Рисунок Б.5 – Прототип «Головної» сторінки з панеллю меню

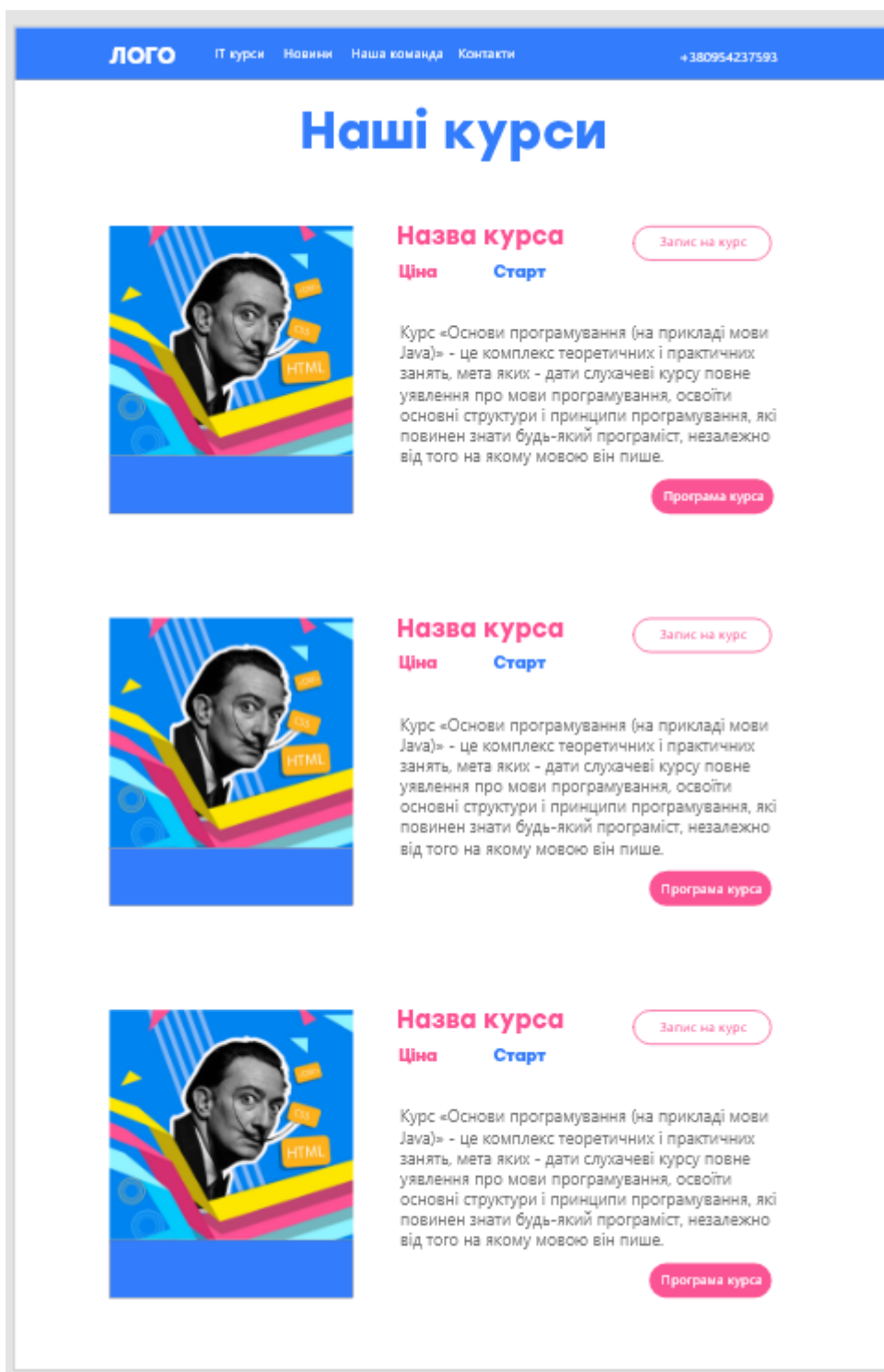


Рисунок Б.6 – Прототип сторінки «Курси»



Рисунок Б.7 – Прототип сторінки «Курси + Запис на курс»



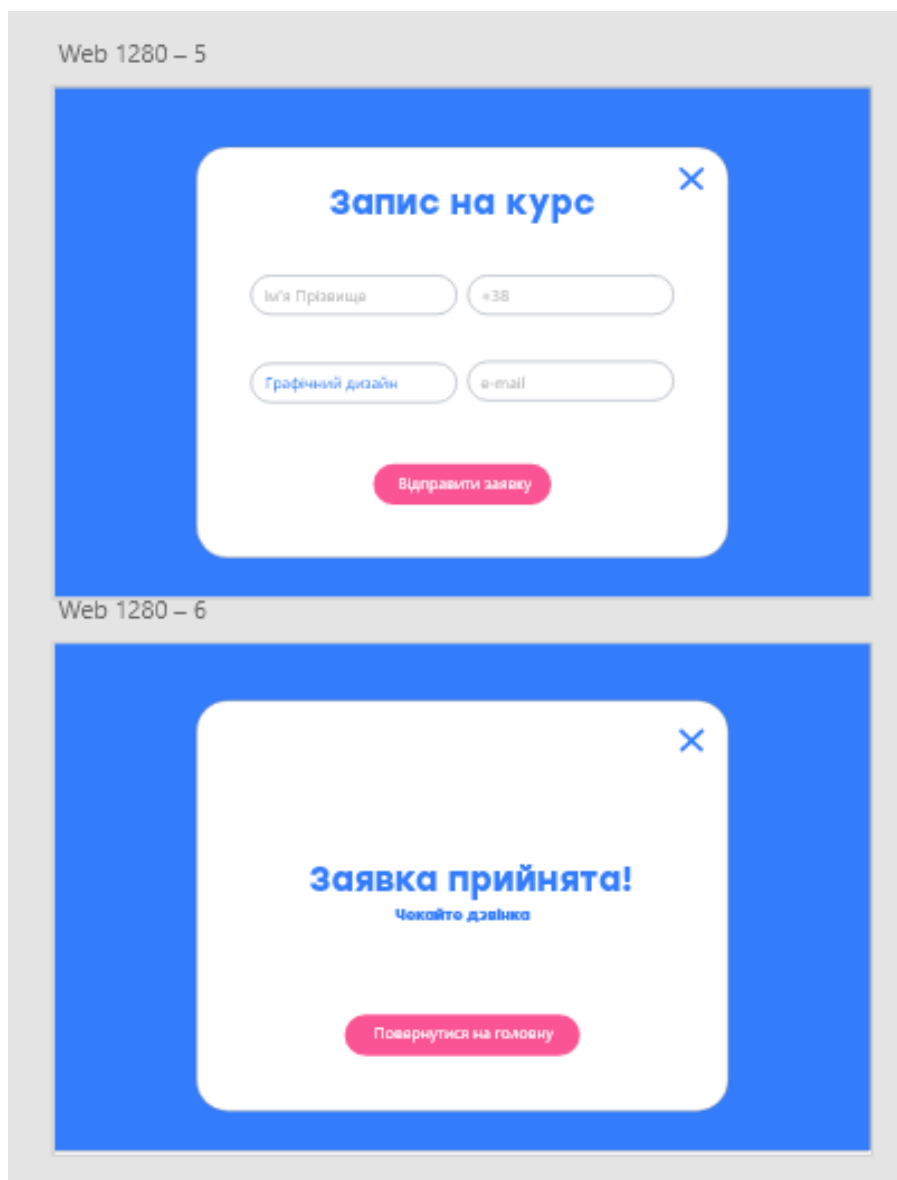


Рисунок Б.8 – Прототип сторінки «Запис на курс», два стани

## 2категорія користувачів. Варіант2

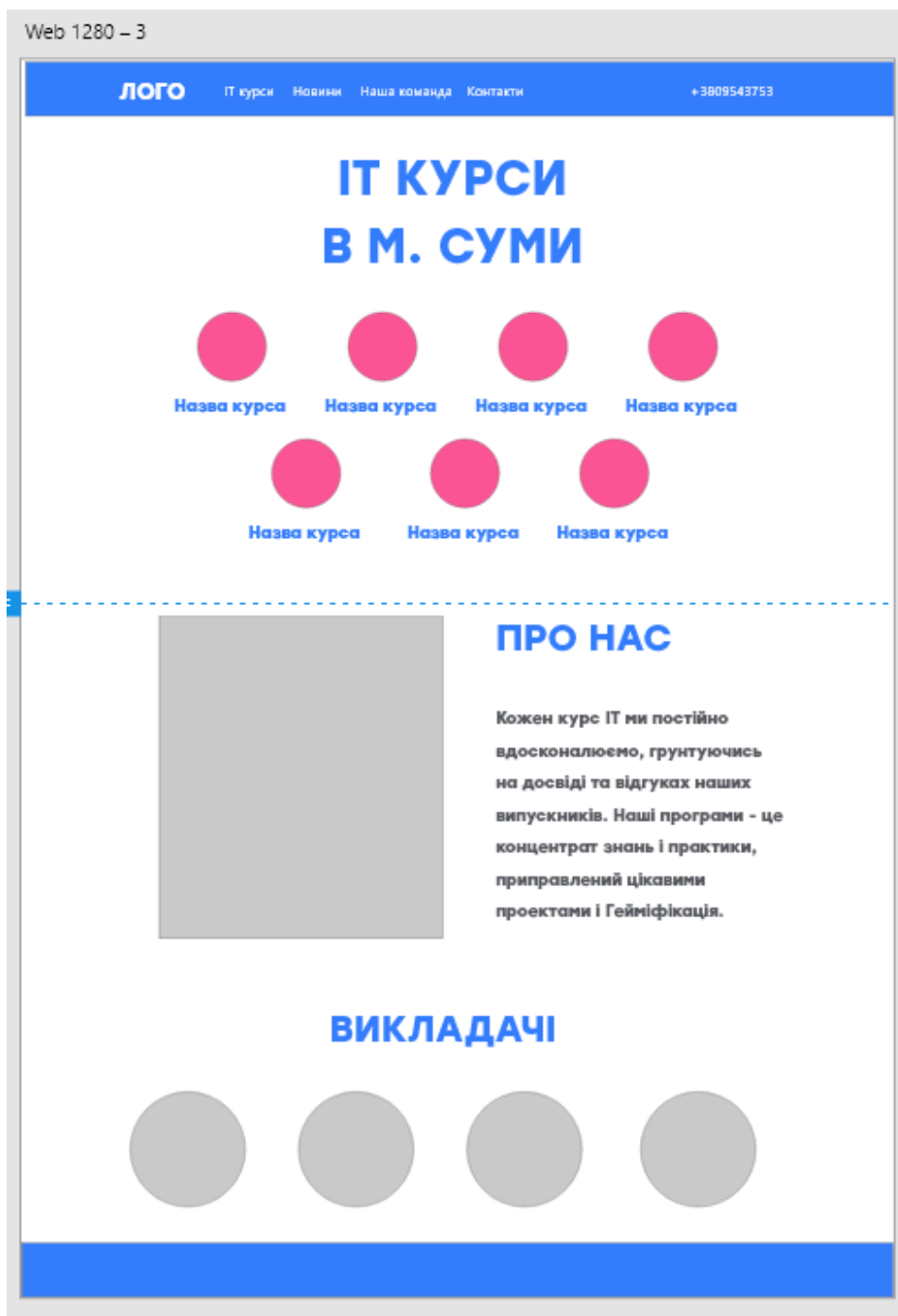



Рисунок Б.9 – Прототип сторінки «Головна»

Web 1280 – 7

ЛОГО П курси Новини Наші освітці Контактти +380954237999

# Назва курсу 1



**Ціна** [Запис на курс](#)

Курс «Основи програмування (на прикладі мови Java)» - це комплекс теоретичних і практичних занять, мета яких - дати слухачеві курсу повне уявлення про мову програмування, оволодіти основні структури і принципи програмування, які повинні знати будь-який програміст, незалежно від того на якій мовою він пише. Цей курс ідеально підійде для тих, хто взагалі нічого не знає і не розуміє в програмуванні, переживає що у нього не вийде, і взагалі ще навіть не визначився

**Складність курсу** ● ● ● ● ○

**Тривалість**

1й місяць

Тема

**Верстка 1. Основи HTML & CSS**

Введення в веб-розробку та інтернет. Розбір основних термінів. Розбір всіх етапів створення сайту. Введення в мову розмітки - HTML: структура документа, ключові слова, заголовки, кодування. Блоків і рядкові елементи. Підія документа.

Види елементів. Специфікатори. Атрибути елементів. Гіперпосилання і їх види. Абсолютні і відносні шляхи. Робота з зображеннями і їх розмірами.

Основи CSS: базовий синтаксис, селектори, класи, спадкування та каскадування.

2й місяць

Тема

**Верстка 1. Основи HTML & CSS**

Введення в веб-розробку та інтернет. Розбір основних термінів. Розбір всіх етапів створення сайту. Введення в мову розмітки - HTML: структура документа, ключові слова, заголовки, кодування. Блоків і рядкові елементи. Підія документа.

Види елементів. Специфікатори. Атрибути елементів. Гіперпосилання і їх види. Абсолютні і відносні шляхи. Робота з зображеннями і їх розмірами.

Основи CSS: базовий синтаксис, селектори, класи, спадкування та каскадування.

[Запис на курс](#)

[Замовити консультацію](#)

## Запис на курс

[Повернутися на головну](#)

Рисунок Б.10 – Прототип сторінки «Курс + Запис на курс»

## ДОДАТОК Г

Копії публікацій за темою:

**« Моделі адаптивного управління людино-машинною  
взаємодією в інформаційних системах»**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

ІНФОРМАТИКА, МАТЕМАТИКА,  
АВТОМАТИКА

**ІМА :: 2019**

**МАТЕРІАЛИ  
та програма**

НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

(Суми, 23–26 квітня 2019 року)

Суми  
Сумський державний університет  
2019

### Метод побудови адаптивного web інтерфейсу для забезпечення usability сайту дизайн студії

Федорова А.В., студент; Лавров Є.А., професор  
Сумський державний університет, м. Суми, Україна

#### Актуальність.

Наростаючий пріоритет використання мобільних пристроїв, а також тенденція підтримки браузером адаптивних ресурсів, призводять до необхідності в адаптивності ресурсів. Однак практика діючих сайтів не враховує всі наростаючі потреби в адаптивності, тому різко знижується юзабіліті цих ресурсів.

#### Постановка задачі.

Розробити та дослідити методику створення адаптивного веб-інтерфейсу, яка збільшить відвідуваність сайту і сприятиме підвищенню економічної ефективності через збільшення кількості клієнтів і результативності їх роботи.

#### Результати.

Відповідно до статистики, кожен третій користувач відвідує інтернет-ресурси з мобільного пристрою, саме тому виникла необхідність у розробці нового адаптивного сайту. Наявні методи не відповідають сучасним вимогам до створення сайтів (приклад методу представлено на рисинку 1):



Рисунок 1 – Існуюча методика

Було проаналізовано наявні підходи до проектування сайтів та визначено, що не всі методи спрямовані на оптимальний результат для цільової аудиторії. Таким чином, існує потреба в розробці ефективної методики проектування адаптивних веб-інтерфейсів, здатна допомогти проектувальникам створювати якісні сайти.

В результаті вивчення об'єкта дослідження, ми можемо висунути гіпотезу про те, що поліпшити методику створення адаптивного інтерфейсу можна:

1. Впровадити тестування протягом всього процесу розробки, що якісно покращить юзабіліті сайту.
2. Застосувати сучасний комплексний підхід, що використовує різні способи верстання, що забезпечує адаптивність.

На рисунку 2 представлена схема поліпшеної методики:



Рисунок 2 – Нова методика

На етапі аналізу ми визначили цільову аудиторію та провели необхідне анкетування, визначили сферу використання сайту.

Прототипування ми створювали на основі сітки «Bootstrap 17», що буде зручно як для упорядкування елементів інтерфейсу, так і для адаптивної верстки.

Дизайн було розроблено для всіх сторінок сайту з врахуванням анкетувань та вибором оптимально зручних рішень.

Розроблений адаптивний сайт також було протестовано командою експертів та пересічними користувачами. За результатами було визначено, що зросла швидкість завантаження сайту, швидкість знаходження потрібної інформації на всіх цільових пристроях та зручність навігації.

#### **Апробація.**

Дану методику було використано для розробки сайту дизайн студії та за оцінками експертів, після впровадження розробленої методики для дизайну, ефективність роботи сайту зросла на 18,5% та збільшилася кількість інтернет замовлень послуг студії.

### Вибір ступеня автоматизації при управлінні технологічним процесом

Федорова А.В., студент; Войцеховский Я., студент;  
Плакс Р., студент; Лавров Є.А., професор  
Сумський державний університет, м. Суми, Україна

Незважаючи на велику кількість досліджень необхідного рівня автоматизації на виробництві задача вибору ступеня автоматизації в інформаційних системах вирішена не до кінця.

Необхідно здійснити розподіл функцій спеціального процесу (алгоритма) функціонування автоматизованої транспортно-складської системи (АТСС) по реалізації заявки гнучкого виробничого модуля (ГПМ) на виконання транспортно-складської операції.

Задача оцінки ступеня автоматизації зводиться до задачі редукції функціональної мережі (ФМ), у якій були виділені наступні функціональні елементи:

- Прийняття заявки на виконання транспортно-складської операції  $f1$
- Визначення адреси комірки складу  $f2$
- Введення завдання в локальну систему управління (ЛСУ) АТСС  $f3$
- Реалізація транспортно-складської операції  $f4$

Аналіз можливостей людини-оператора і машини з виконання виділених операцій з врахуванням того, що основні технічні засоби задані і зміни ступеня автоматизації можливі за рахунок введення програмної реалізації деяких операцій, що дозволить сформулювати множини можливих способів виконання кожної операції.

Для того щоб, побудувати модель алгоритма функціонування ЛМС, було визначено типи функціонерів, які необхідно поставити у відповідність виділеним функціональним елементам:

- $f1$ - робоча ТФО (позначимо  $P1$ );
- $f2$ - робоча ТФО (позначимо  $P2$ );
- $f3$ - робоча ТФО (позначимо  $P3$ );
- $f4$ - робоча операція з самоконтролем працездатності (позначимо  $RKR$ , Контроль працездатності введемо в зв'язку з великою ймовірністю відмови штабелера та транспортного візка);
- $f_1^k$ - контрольна ТФО (позначимо  $K1$ );
- $f_2^k$ - контрольна ТФО (позначимо  $K2$ );



Окрім введених основних і контрольних операцій в модель було введено операцію, яка відповідає ремонту технологічного обладнання  $f_1^P$ . Після цього було побудовано алгоритм(рис. 1)

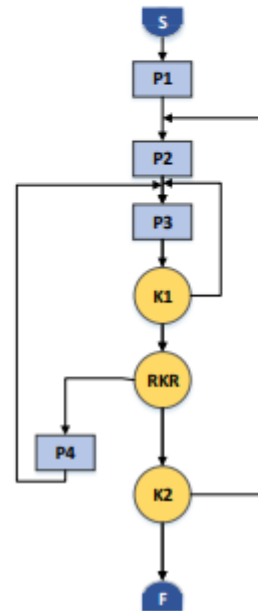


Рисунок 1 – Алгоритм роботи оператора

Після аналізу різних даних для цього алгоритма у нас були різні комбінації вхідних характеристик, які було промодельовано у розробленому програмному комплексі. Було обрано оптимальний варіант, який мав найвищі значення ймовірності своєчасного виконання алгоритма та ймовірності безпомилкового виконання

Даний метод було використано на виробництві при прийнятті рішення про вибір оптимального ступеня автоматизації технологічного процесу.



## ДИАЛОГОВЫЙ МОДЕЛИРУЮЩИЙ КВАЛИМЕТРИЧЕСКИЙ КОМПЛЕКС ДЛЯ ЭРГОНОМИЧЕСКОГО ОБЕСПЕЧЕНИЯ ЦИФРОВЫХ ТЕХНОЛОГИЙ УПРАВЛЕНИЯ

Е. А. Лавров, Н. Б. Пасько, А. В. Федорова, Е. Плеханов

Сумский государственный университет

Сумы

*prof\_lavrov@mail.ru*

Проанализированы задачи эргономического обеспечения цифрового управления. Разработана информационная технология моделирования и оптимизации человеко-машинного взаимодействия в системах цифрового управления. Охарактеризована сфера применения информационной технологии.

**Ключевые слова:** цифровая технология управления, эргономика, человек-оператор, деятельность, моделирование, квалиметрия, оптимизация.

## DIALOGUE MODELING KVALIMETRIC COMPLEX FOR ERGONOMIC SUPPORT OF DIGITAL MANAGEMENT TECHNOLOGIE

E. A. Lavrov, N. B. Pasiko, A. V. Fedorova, E. Plehanov

Sumy State University

Sumy

The tasks of ergonomic provision of digital control are analyzed. The information technology of modeling and optimization of human-machine interaction in digital control systems has been developed. The scope of information technology is characterized.

**Key words:** digital control technology, ergonomics, human operator, activity, modeling, qualimetry, optimization.

**Введение.** От 50 до 80% аварий в системах цифрового управления разных типов, более 64% катастроф на морском флоте и 80% в авиации вызваны ошибками человека-оператора. Исследования в области проектирования человеко-машинных систем (ЧМС) ставят целью уменьшить ошибочные реакции человека-оператора, обеспечить допустимые условия труда, уменьшить травматизм и стрессы людей, приспособить «машинную часть» к человеку [1-3]. Эффективность эргономических исследований существенно зависит от возможности формализовать взаимодействие человека с машиной и получать оперативные оценки вариантов организации деятельности, решать задачи оптимизации технологий цифрового управления.

**Постановка задачи.** Разработать интерактивный моделирующий квалиметрический комплекс для эргономического обеспечения цифровых технологий.

**Результаты.** В качестве базовой выбрана методология функционально-структурной теории эрготехнических систем проф. А.И. Губинского.

При обосновании структуры системы обеспечения эргономического качества цифровых информационных систем выделяем комплексы работ, которые проводятся на этапах: «Проектирование»; «Изготовление»; «Эксплуатация».

Особенностью современных ЧМС являются возрастающие полиэргатичность, гибкость и необходимость оперативной «перестройки» (закрепления функций «под задачу»).

Поэтому многие задачи, которые решались ранее 1 раз на этапе проектирования и использовались многократно при эксплуатации системы, для современных полиэргатических гибких систем с нерегламентными задачами должны решаться оперативно с учетом большого количества особенностей и ограничений.

Для обеспечения оперативного компьютерного моделирования получены новые математические модели:

- системного анализа АСУ (компонентные и морфологические)
- языка описания дискретного человеко-машинного взаимодействия
- автоматической редукции функциональной сети, описывающей алгоритм функционирования ЧМС
- вывода расчетных зависимостей, необходимых для оценивания типовых функциональных структур, используемых при редукции функциональной сети (в том числе с учетом ошибок разных типов)
- выбора оптимального варианта человеко-машинного взаимодействия (в разных постановках, для разных задач эргономического обеспечения – определение степени автоматизации, расчет количества операторов, выбор структур деятельности и способов выполнения операций, распределения функций между операторами, определения мероприятий системы обеспечения эргономического качества и др.
- эргономической экспертизы элементов автоматизированных технологий [4]

Реализована компьютерная технология, позволяющая в диалоговом режиме моделировать ЧМС:

- описывать
- оценивать
- оптимизировать

взаимодействие операторов с программно-техническими средствами АСУ, а также вести необходимые модели «человека», техники, справочники о надежности и качестве выполнения типовых действий и операций. Оптимизация человеко-машинного взаимодействия предусматривает возможность изменять технологии функционирования таким образом, чтобы реализовать адаптацию автоматике к особенностям человека-оператора.

**Апробация.** Информационная технология прошла апробацию при проектировании и эксплуатации автоматизированных систем различного назначения:

- контакт-центров [5]
- систем предоставления доступа к интернет-ресурсам[6]
- гибких производственных систем[7]
- систем управления логистическими компаниями[8-9]
- управления магистральным газопроводом[10-11]
- расчетных центров[12]
- e-learning [13]
- производственных процессов машиностроительных предприятий [14-15].

- предприятий химической промышленности [16].
- и др.

**Выводы.** Информационная технология является практическим результатом многолетних научных исследований по моделированию и оптимизации человеко-машинного взаимодействия в рамках научной школы «Эффективность, качество и надежность эргономических систем» проф. А.И. Губинского и может быть полезна при создании и внедрении мероприятий по обеспечению эргономического качества цифровых технологий в образовании, науке, обществе.

#### Библиографический список

1. Rothmorea, P., Aylwardb, P., Karnona J. The implementation of ergonomics advice and the stage of change approach [Text]. / P. Rothmorea, P. Aylwardb, J. Karnona // *Applied Ergonomics*. – 2015. – № 51. – P. 370–376.
2. Bentley, T.A., Teo, S.T.T., McLeod, L., Tana, F., Bosua, R., Gloet, M. The role of organisational support in teleworker wellbeing: A socio-technical systems approach [Text] / T.A. Bentley, S.T.T. Teo, L. McLeod, F. Tana, R. Bosua, M. Gloet // *Applied Ergonomics*. – 2016. – № 52. – P. 207–215.
3. Информационно-управляющие человеко-машинные системы: исследование, проектирование, испытания: Справочник / Под общ. ред. А.И. Губинского и В.Г. Еаграфова. – М.: Машиностроение, 1993. – 528с.
4. Lavrov, E. Organizational Approach to the Ergonomic Examination of E-Learning Modules [Text] / E. Lavrov, O. Kurenko, T. Lavryk, N. Barchenko // *Informatics in education*, 2013. –V. 12. –N. 1. –P105–123.
5. Lavrov, E. Modelling Of Operator's Activity In Contact Center Of Providing Internet And Television Services[Text] / E. Lavrov, A. Krivodub, Y. Shapochka // *International Scientific Conference «UNITECH '16». Proceedings. 18-19 November 2016, Gabrovo, Bulgaria*. - Gabrovo: University Publishing House «V. APRILOV», 2016. – Volume 2. - P.p 195-200.
6. Криводуб А.С. Оценка надежности деятельности операторов в системах предоставления доступа к ресурсам компьютерных сетей // *Вісник Національного технічного університету «ХПІ»*. Серія: Нові рішення у сучасних технологіях. – 2016. – № 18 (1190). – С.140-147.
7. Lavrov, E. Mathematical models for the distribution of functions between the operators of the computer-integrated flexible manufacturing systems / N. Pasko, A. Krivodub, A. Tolbatov // *proceedings of the XIII-th international conference tcset'2016 «modern problems of radio engineering, telecommunications, and computer science»*. – Lviv-Slavsko, Ukraine, february 23 – 26, 2016. – p. 72-76.
8. Lavrov, E. Ergonomics of IT outsourcing. Development of a mathematical model to distribute functions among operators [Text] / E. Lavrov, N. Pasko, A. Krivodub, N. Barchenko, V. Kontsevich // *Eastern European Journal of Enterprise Technologies*. 2016. – N.4 (80). – P. 32-40.
9. Лавров Е.А., Криводуб А.С. Подход к оценке вариантов деятельности операторов технической поддержки информационных сервисов телекоммуникационных систем // *Доклады БГУИР*. – Минск, 2015. - №2 (88). - С. 123-126.
10. Кошара В.С., Лавров Е.А. Формализованное описание деятельности операторов системы управления газоперекачивающей установкой // *Інформатика, математика, автоматика: матеріали та програма науково-технічної конференції, м. Суми, 18-22 квітня 2016 р.* – Суми: СумДУ, 2016. – С. 96.

11. Koshara V., Krivodub A., Pasko, N., Lavrov E. Information Technology Distribution of Applications between Operators of the Compressor Station //Advanced Information Systems and Technologies: proceedings of the IV international scientific conference, May 25-27, 2016 - Sumy: Sumy State University, 2016. - P. 89.
12. Lavrov, E. Information technology for distribution of functions between operators in automated systems. Analysis of efficiency. [Text] / E. Lavrov, N. Pasko, // International Scientific Conference «UNITECH '15». Proceedings. 18-19 November 2015, Gabrovo, Bulgaria. - Gabrovo: University Publishing House «V.APRILOV», 2015. – Volume 2. - P.p 298-306.
13. Lavrov E. Development of models for the formalized description of modular e-learning systems for the problems on providing ergonomic quality of human-computer interaction/ E Lavrov, N Barchenko, N Pasko, I Boroznec// Eastern-European Journal of Enterprise Technologies 2 (2 (86)), 4–13.
14. Bahmach M., Lavrov E. Program Complex of Statistical Calculations for Control the Quality of Products at Lebedinsky Plant of Piston Rings Advanced Information Systems and Technologies: proceedings of the IV international scientific conference, May 25-27, 2016– Sumy: Sumy State University, 2016. – P. 82-84.
15. Бахмач Н.В., Лавров Е.А. Формализованное описание производственных процессов на Лебединском заводе поршневых колец для задач управления качеством // Информатика, математика, автоматика: матеріали та програма науково-технічної конференції, м. Суми, 18-22 квітня 2016 р. – Суми : СумДУ, 2016. – С. 90.
16. Лавров Е.А., Схидаченко А.С. Эргономические резервы повышения эффективности АСУТП производства удобрений //Сучасні інформаційні системи і технології: Матеріали Другої міжнародної науково-практичної конференції, м. Суми, 21-24 травня 2013 р.— Суми : СумДУ, 2013. — С. 53-54.

## СОВЕРШЕНСТВОВАНИЕ ЦИФРОВЫХ ТЕХНОЛОГИЙ ПРОИЗВОДСТВА МЕТОДАМИ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ ЧЕЛОВЕКО-МАШИНЫМ ВЗАИМОДЕЙСТВИЕМ

Е. А. Лавров, Н. Б. Пасько, Т. В. Щербань, Ю. С. Михайленко

Сумський державний університет

Суми

[prof\\_lavrov@mail.ru](mailto:prof_lavrov@mail.ru)

Проанализированы эргономические проблемы современного цифрового управления. Предложен метод оптимизации алгоритма деятельности человека-оператора. Охарактеризована сфера возможных применений метода. Предложены пути широкого внедрения методов оптимизации в практику эргономического обеспечения.

**Ключевые слова:** цифровая технология производства, эргономика, человек-оператор, деятельность, оптимизация, надежность.

**TECHNICAL UNIVERSITY OF GABROVO**



**INTERNATIONAL  
SCIENTIFIC CONFERENCE**

**UNITECH 2017  
GABROVO**

**P R O G R A M**

**17 - 18 NOVEMBER 2017  
GABROVO**

**METHOD OF FORMALIZED DESCRIPTION OF INTERACTION IN  
MAN-MACHINE SYSTEMS**

**МЕТОД ФОРМАЛИЗИРОВАННОГО ОПИСАНИЯ ВЗАИМОДЕЙСТВИЯ В  
ЧЕЛОВЕКО-МАШИННЫХ СИСТЕМАХ.**

**Pasko N.B.**

*Sumy National Agrarian University (Ukraine)*

**Lavrov E.A.**

**Fedorova A.V**

*Sumy State University (Ukraine)*

**Abstract**

*The task of formalization of description of functional network of algorithm of activity of operator of the systems of «man-machine» is examined. The method of description of functional networks is developed.*

**Keywords:** algorithm of functioning, model of functional network, model of functional units, model of functional structures.

*INTRODUCTION*

The effectiveness of automated systems depends significantly on the consideration of the so-called "human factor" [1-2]. Creation by Professor Gubinsky AI scientific school "The effectiveness, quality and reliability of systems" man - technology - environment "was in fact a revolution in ergonomics, which allowed to find approaches to the formalization and optimization of the human operator [1-4]. The activity of the human operator [1,3,4] is specified by the functional network (FN), but

there are big problems associated with the convenience of entering the structure of this FN into the computer for the purposes of the subsequent simulation of human-machine interaction.

The purpose of this work is to develop an approach [2,5] to the development of a language describing the FN, which allows:

- describe (in a form convenient for entering into the computer) FN including:

- operations algorithm of operation - typical functional units (TFU);



- communication between operations (TFU);
- characteristics of operations;
- typical blocks of operations - type functional structures (TFS);
- rules for identifying TFS;
- rules for converting (reducing) the FS;
- models for calculating the indicators at the steps of reduction;
- to ensure automatic reduction and evaluation indicators of FS.

*exposition*

### 1. Development of models for describing and evaluating FS in general.

By a formalized description of a functional network, we mean the representation of the algorithm for the functioning of the human machine system (HMS) by a structural formula consisting of elements of two sets  $M_1$  and  $M_2$ :

$M_1$  – a set of elements of the description of FN, with the help of which a description of the functional network of the activity algorithm is constructed, taking into account events leading to the generation, detection and elimination of errors (including various types);

$M_2$  – a set of types of links (description operations) between elements of the description of the FS. The elements of this set are relations between the elements of the description of a particular FN (order, outputs and transitions).

Then we obtain a model of FN description in general form:

$$O_{FS} ::= \langle E_{FS}, S_{FS} \rangle, \quad (1)$$

where:  $E_{FS}$  – a subset of description elements from the set  $M_1$ ,  $E_{FS} \in M_1$ ;  $S_{FS}$  – a subset of operations describing the set  $M_2$ ,  $S_{FS} \in M_2$ .

The process of estimating a functional network is also given by elements of two sets  $M_3$  and  $M_4$ :

$M_3$  – a set of elements of evaluation; probabilistic and cost characteristics, with the help of which the quality of the performance of the elements of the description of the FS is assessed;

$M_4$  – a set of estimation operations, that is, operations on the evaluation elements, by

means of which the probability-time characteristics of the entire algorithm of the HMS operation are calculated. To the set  $M_4$  we refer the library of known mathematical models for calculating the performance indicators of typical functional structures and replacing them by equivalent standard functional units with one error in mind and developed models for accounting for errors of various types. We introduce a one-to-one correspondence: first, between the elements of the description of  $M_1$  and the elements of the estimation from  $M_3$ ; secondly, between the operations of the description of  $M_2$  and the operations of estimation from  $M_4$ . This makes it possible to quantify the algorithm of the HMS operation by a formal procedure: it is sufficient to describe the operation algorithm by the sets  $M_1$  and  $M_2$ , and then to identify and replace each description operation from the set  $M_2$  by the corresponding estimated operation from the set  $M_4$ . The procedure for replacing the description operations with their estimated analogs in the study of a formalized FN will be called the folding or reduction of FN (by analogy with the reduction of the graphical representation of FN). This procedure is based on the use of typical functional units and structures.

Then we obtain a model of FN estimation in general form:

$$C_{FS} = \langle C_{EFS}, C_{OFS}, Pr_{FS} \rangle, \quad (2)$$

where:  $C_{EFS}$  – a subset of the estimation elements from the set  $M_3$ ,  $C_{EFS} \in M_3$ ;  $C_{OFS}$  – a subset of estimation operations from the set  $M_4$ ,  $C_{OFS} \in M_4$ ;  $Pr_{FS}$  – reduction protocol model of FN.

Define the values of the description elements, evaluation elements and description operations used to represent the functional network and other objects.

**Description of FN elements.** With the help of the description elements (elements of the set  $M_1$ ), the designations of the TFU (the functionaries of the basic and additional and composers of auxiliary and service), the name of TFU and the name of equivalent TFU in the structure of the activity algorithm. Functionaries correspond to real operations or actions of a person, work operations of

technological equipment, computer hardware and software in the analyzed algorithm of functioning, and composes to certain interrelationships of operations and logical functions. The designations of some units of functioning, taken to denote the elements of the FN, their names are given in Table 2.

Table 2 - Elements of the description for designation of the basic typical units of functioning

Number	Description element definition	Application for description
1	R	Working operation
2	A	Alternative operation
3	K	Operation control function
4	Z	Delay operation

As the description elements for specifying the name of TFU and equivalent TFU in the structure of the activity algorithm, we will use symbol sequences that indicate the type of TFU or equivalent TFU and the number of this TFU in the structure of the algorithm:

«P1», «P2»,..., «Pn», «K1», «K2»,..., «Pe1»,... .

**The set of elements of estimation.**

Elements of this group are used as variables for setting values of quality indicators of descriptive elements from the set  $M_1$ . A variable, like in mathematics, is an object with a name and meaning. The name is used to indicate the quality score. The variable gets a specific value in the process of setting the value to the evaluation elements (elements of the set  $M_3$ ), i.e. probabilistic and temporal characteristics, with the help of which the quality of performance of descriptive elements is evaluated. Examples of variables and their use are shown in Table 3

**Model of TFU.** A formalized description of a functional unit is an element of the description from the set  $M_1$ , together with the corresponding elements of the evaluation from the set  $M_3$ .

Then the model of the  $i$ -th model functional unit will look like:

$$Fe_i ::= \langle o_{e_i}, \{c_{e_{ij}}\} | j = 1, 2, \dots, m_i \rangle, \tag{3}$$

$$o_{e_i} \in M_1, c_{e_{ij}} \in M_3$$

where:  $O_{e_i}$  - the  $i$ -th element of the description of the set  $M_1$ ;  $C_{e_{ij}}$  -  $j$ -th evaluation element ( $j$ -th quality indicator) of the  $i$ -th item from the set  $M_1$ ;  $m_i$  - the number of evaluation elements related to the  $i$ -th descriptive element.

Table 3 - Evaluation elements

Number	Designation of the evaluation element	Application for setting values
1	$B^1$	The probability of error-free execution of TFU
2	$B^0$	Probability of erroneous execution of TFU
5	$M_T$	Expected value of execution time
6	$D_T$	Dispersion of execution time
7	$K^{11}$	The conditional probability that the operation under test, if actually performed correctly, will be recognized as correct
8	$K^{00}$	The conditional probability that the operation to be checked, if actually performed incorrectly, will be recognized as incorrect

Using the obtained model of the functional unit in general form (formula 3), as well as the elements of description and estimation defined above, we will describe some functional units (Table 4).

Table 4 - Description of the functional units

Number of TFU	Contents of TFU	Model of TFU
1	Working	$Fe_1 = \langle R, B^1, B^0, [M_T, D_T], [\bar{C}, D(C)], [\bar{W}, D(W)] \rangle$
2	Alternative	$Fe_2 = \langle A, A_i, A_{ij}, [M_T, D_T], [\bar{C}, D(C)], [\bar{W}, D(W)] \rangle$

3	Function monitoring	$Fe_3 = \langle K, K^{11}, K^{10}, K^{00}, K^{01}, [M_T, D_T], [\bar{C}, D(C)], [\bar{W}, D(W)] \rangle$
---	---------------------	--

**Description of the relationships between the elements of the FN.** The connections between the elements of the FN are specified by the elements of the set  $M_2$ , that map how connected and in which sequence the units of operation are performed. Elements of this group are generally denoted by  $N_j, V_{j_i}, L_{j_i}$  variables with lower indices, used in models to define relationships between elements of functional structures and a functional network, and take integer values. The use of these variables:

$N_j$  – the serial number of TFU in the structure of the algorithm;  $V_{j_i}$  – the possible type of transition after the execution of the TFU with the number  $N_j$ ;  $L_{j_i}$  – number of the TFE executed after the execution of the TFE with the number  $N_j$ , if a type transition occurs.  $V_{j_i}$ .

The list of possible types of transitions and the corresponding values of the variable  $V_{j_i}$  are given in Table 5.

Table 5 - Application of the variable  $V_{j_i}$

$$Fs_i ::= \langle \{o_{e_{ij}}, N_{ij}, \{V_{ij_l}, L_{ij_l}\} | l=1,2,\dots,\eta_{ij}, [k_{c_j}]\} | j=1,2,\dots,k_i, o_{ee_i}, \{y_{i_m}\} | m=1,2,\dots,z_i \rangle \quad (4)$$

where:  $o_{e_{ij}}$  –  $j$ -th element of the description of the  $i$ -th functional structure;  $k_i$  – number of elements of the description of the  $i$ -th functional structure;  $k_{c_j}$  – restriction on the number of repetitions in a cycle;

$\{N_{ij}, \{V_{ij_l}, L_{ij_l}\} | l=1,2,\dots,\eta_{ij}\} \in M_2$  – a subset of links (description operation) corresponding to the  $j$ th element of the description of the  $i$ -th functional structure. Denotes the transition  $V_{ij_l}$  from the functional

unit with the descriptive element  $o_{e_{ij}}$  and the number  $N_{ij}$ , to the functional unit with the number  $L_{ij_l}$ ;  $\eta_{ij}$  – number of types of

Value of variable $V_{j_i}$	Use to specify the type of transition
1	The transition to TFU in the structure of the algorithm following the given in the main direction of the algorithm
2	Transition to the TFU, which follows the current TFU "operation control" in case of erroneous execution of the operation
3	Transition to the TFU following the current TFU "performance check" in case of erroneous execution of the operation
4	Transition to the TFU of the continuation of the cycle following the current TFU
5	The transition to the TFU, which follows the current TFU and signifies the exit from the cycle

In the group of elements of the assignment of links we add the variable  $kc$ , which means the restriction on the number of repetitions in the cycle.

**Model of TFS.** A formalized description of the functional structure is a description of the operation from the set  $M_2$ , that defines the relations between several elements of the description, together with the corresponding valuation operations from the set  $M_4$ . Then the model of the  $i$ -th model functional structure will look like:

transitions corresponding to the  $j$ -th descriptive element of the  $i$ -th functional structure;

$o_{ee_i}$  – element of the description of the equivalent functional unit of the  $i$ -th functional structure;

$\{y_{i_m}\}$  –  $m$  – the valuation operation of the  $i$ -th functional structure used to determine the  $m$ -th quality index of equivalent TFU,  $\{y_{i_m}\} \in M_4$

;  $z_i$  – number of evaluation operations of the  $i$ -th functional structure.

Using a model of the functional structure (formula 4) defined in general form, as well as the descriptive, estimated elements and elements of the task of descriptive operations defined above, we give a description of the

functional structures (examples of the TFS models are Table 6))

*Table 6 - Examples of the description of functional structures*

Number of TFS	Designation of TFS	TFS content	Model of TFS
	$FS_{RR}$	Consistent execution of work operations	$FS_{RR}$ = $\langle \{R,1,(1,2)\}, \{R,2,(1,3)\}, \dots, \{R,n,(1,n+1)\}, R, \{B,MT,DT\} \rangle$
2	$FS_{RK}$	Cyclic FN "Working operation with monitoring of operation without limit on the number of cycles"	$FS_{RK} = \langle \{R,1,(1,2)\}, \{K,2,(1,3),(2,1)\}, R, \{B,MT,DT\} \rangle$
3	$FS_{CRF}$	$n$ -fold repetition of the work operation with acceptance for all successful outcomes	$FS_{CRF}$ = $\langle \{R,1,(1,2)\}, \{C_F,2,(4,1),(5,3),k_c\}, R, \{B,MT,DT\} \rangle$

**FN model of operator activity algorithm.** Taking into account the description elements and description operations introduced in the models (3) and (4), the structural formula (1) of the functional network representation by elements of the sets  $M_1$  and  $M_2$  will look like:

$$O_{FS} ::= \langle \{o_{e_j}, te_j, N_j, \{V_{j_l}, L_{j_l}\} \\ | l = 1, 2, \dots, \eta_j, [k_{c_j}]\} \mid j = 1, 2, \dots, n \rangle, \quad (5)$$

where:  $o_{e_j}$  -  $j$ -th element of the description in the structure of the algorithm of activity;  $te_j$  – the designation in the structure of the algorithm of the functional unit  $c$  with the description element  $o_{e_j}$ ;  $n$  - the number of elements of the description in the structure of the algorithm;  $N_j, \{V_{j_l}, L_{j_l}\}$  - descriptive operation corresponding to the  $j$ -th element of the functional network description. Denotes a type-  $V_{j_l}$  transition from a functional unit to a description element  $o_{e_j}$  and number  $N_j$ , to the functional unit with the number  $L_{j_l}$ ;  $\eta_j$  – number of transition types corresponding to the  $j$ -th element of the description;  $k_{c_j}$  - restriction on the number of repetitions in a cycle.

#### CONCLUSION

The developed models were used in the creation of computer simulation technology of human-machine interaction [6-7], which confirms the constructiveness and convenience of the approach.

#### REFERENCES:

- [1] Информационно-управляющие человеко-машинные системы: исследование, проектирование, испытания: Справочник / Под общ. ред. А.И. Губинского и В.Г. Евграфова. – М.: Машиностроение, 1993. – 528с.
- [2] Lavrov E. Computer Simulation of Systems “Man-Machine”: Achievements and Tasks // Materials International Scientific Conference “UNITECH ‘07”. – Gabrovo, Bulgaria. – 2007. – Vol. 3. – P. 358–362.
- [3] Попович П.Р. Эргономическое обеспечение деятельности космонавтов /П.Р. Попович, А.И. Губинский, Г.М. Колесников.-М.: Машиностроение, 1985.-272с.
- [4] Чабаненко П.П. Исследование безопасности и эффективности функционирования систем “человек-техника” эргосетями: Севастополь, АВМС им. П. С. Нахимова, 2012.-162с.
- [5] Лавров, Е.А. Язык описания функциональных сетей для моделирующего квалиметрического комплекса эрготехнических систем / Е.А. Лавров, Н.Б. Пасько // Восточно-Европейский журнал передовых технологий. Сер. Информационно-управляющие системы. - Харьков, 2010 - 3/11 (45) С.4-10.
- [6] Лавров, Е.А. Компьютерная технология моделирования дискретного человеко-машинного взаимодействия / Е.А. Лавров, Н.Б. Пасько, Б.В. Панов // Научная мысль. – 2014. - №1. – С. 48 – 60.
- [7] Лавров Є.А., Пасько Н.Б. Свідство про реєстрацію авторського права на твір «Комп'ютерна програма «Комп'ютерна технологія моделювання дискретної людино-машинної взаємодії», № 45262, 21.08.2012.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

ІНФОРМАТИКА, МАТЕМАТИКА,  
АВТОМАТИКА

**ІМА :: 2018**

**МАТЕРІАЛИ  
та програма**

**НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ**

(Суми, 5-9 лютого 2017 року)



Суми  
Сумський державний університет  
2018

**Метод автоматической редукции функциональных сетей для задач моделирования операторской деятельности в критических системах**

Федорова А.В., *студентка*

Лавров Е.А., *профессор*

Сумский государственный университет, г. Сумы, Украина

Пасько Н.Б., *доцент*

Сумский национальный аграрный университет, г. Сумы, Украина

**Введение**

Комплексная автоматизация производства не только не освобождает человека от оперативного управления технологическими процессами, но и усложняет его деятельность. По вине человека происходит более 80 процентов аварий в т.н. критических системах. Это - системы с большими ущербами, которые могут возникнуть вследствие различных нарушений (химическое производство, энергетика, магистральные газопроводы, авиация и т.п.).

**Постановка задачи.**

С целью проектирования оптимальных алгоритмов деятельности в условиях жестких временных ограничений для принятия решений обеспечить возможность оперативного оценивания надежности операторской деятельности. Для этого необходим метод автоматического (без участия человека) оценки вариантов возможных действий.

**Результаты.**

Задача оценивания сведена к задаче редукции функциональной сети (ФС), описывающей деятельность оператора. ФС задается с помощью процедуры установления логико-временных связей между операциями или типовыми функциональными единицами деятельности (ТФЕ). Для типовых блоков операций, т.н. типовых функциональных структур (ТФС) выведены математические модели для оценки показателей безошибочности и времени выполнения. При этом основная сложность состоит в обеспечении идентификации в структуре ФС типовых блоков (ТФС) и реализации процедуры автоматической редукции ("сворачивания") ФС.

Для проведения редукции разработаны :

- библиотека описаний ТФС;
- технология автоматической редукции ФС ( Рис.1)

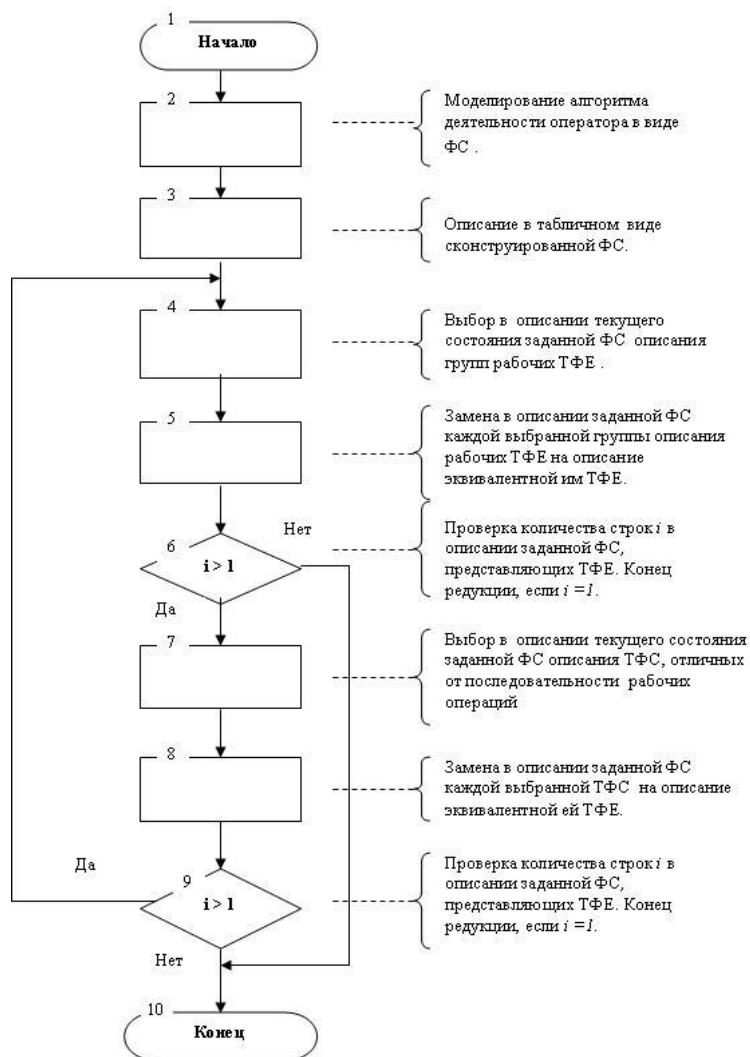


Рисунок 1- Основные этапы процесса редукции ФС

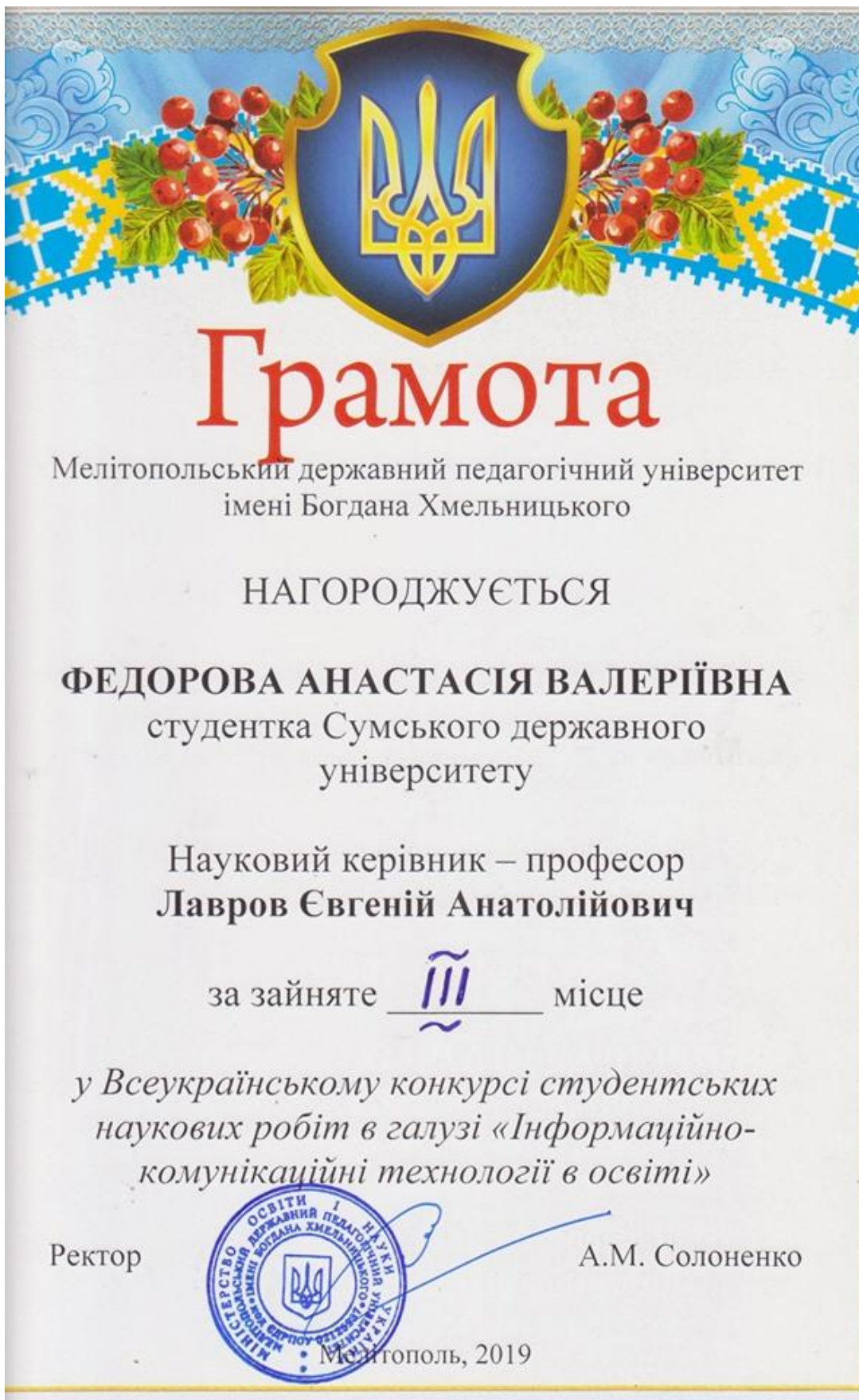
**Апробация**. Метод использован при разработке моделирующего квалитметрического комплекса эрготехнических систем.



## ДОДАТОК Д

Грамоти отримані у всеукраїнських конкурсах студентських наукових робіт





# Грамота

Мелітопольський державний педагогічний університет  
імені Богдана Хмельницького

НАГОРОДЖУЄТЬСЯ

**ФЕДОРОВА АНАСТАСІЯ ВАЛЕРІЇВНА**  
студентка Сумського державного  
університету

Науковий керівник – професор  
**Лавров Євгеній Анатолійович**

за зайняте III місце

*у Всеукраїнському конкурсі студентських  
наукових робіт в галузі «Інформаційно-  
комунікаційні технології в освіті»*

Ректор



А.М. Солоненко

Мелітополь, 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ



# ДИПЛОМ

НАГОРОДЖУЄТЬСЯ

Кіншаков Едуард Віталійович  
Федорова Анастасія Валеріївна  
Сумський державний університет

за III місце

у II турі Всеукраїнського конкурсу студентських  
наукових робіт 2018/2019 навчального року  
зі спеціальності  
*«Автоматизація та  
комп'ютерно-інтегровані технології»*

Голова галузевої  
конкурсної комісії  
25 квітня 2019 р.



І.В. Рубан



ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ

АВТОМОБІЛЬНО-ДОРОЖНИЙ

**ДИПЛОМ**

нагороджується

ДИПЛОМОМ **III** СТУПЕНЯ

студентка

Сумського державного університету

*Федорова Анастасія Валеріївна*

за  
наукову роботу у Всеукраїнському конкурсі  
студентських наукових робіт з

19 квітня 2019 р.

Ректор



А.М.Туренко