

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса  
Шевченка ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
**Кафедра програмних систем і технологій**

Дисципліна  
**«Архітектура та проектування програмного забезпечення»**

**Лабораторна робота № 4**  
**«Аналіз архітектури програмного забезпечення»**

**на тему:**  
**«Онлайн клієнт для погодних даних»**

Виконав:	Гоша Давід	Перевірів:	Берестов Д.С
Група	ІПЗ-23	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		
2022			

**Мета практикуму** – отримати практичні навички підготовки та проведення архітектурного аналізу.

**Тема проекту** – «Онлайн клієнт для погодних даних».

**Застосування проекту** - проект розробляється та проектується з метою полегшення роботи користувача з великим обсягом даних.

**Опис завдань та обсягу проведення архітектурного аналізу:**

Спочатку визначимося з важливістю правильного вибору архітектурного шаблону, зрозуміємо, що таке архітектура, архітектурний аналіз та для чого вони потрібні.

**Архітектура** - це особливий рівень проекту: "Окрім створення алгоритмів і структур даних, необхідно вирішити ще одне принципове завдання - розробити загальну структуру системи. Процес розробки структури включає створення загальної інфраструктури організації системи та управління нею, вибір протоколів і методів синхронізації і доступу до даних, розподіл функцій системи між компонентами, фізичний розподіл, об'єднання елементів проекту, масштабування, оптимізацію продуктивності та вибір оптимальних варіантів серед доступних альтернатив.

**Архітектурні уявлення** охоплюють основні рішення, прийняті під час виборів структури програмного забезпечення, і демонструють декомпозицію архітектури складові її компоненти, з'єднувачі і форми. Рішення, що приймаються при виборі структури, обумовлені функціональними та додатковими вимогами та іншими обмеженнями. У свою чергу, ці рішення породжують нові обмеження щодо **вимог** та подальших рішень на нижчих рівнях.

**Архітектурні вимоги** - це "умова чи можливість, яким має відповідати система". Виділимо ряд вимог. Існує багато різновидів вимог. Одна із класифікацій вимог називається моделлю FURPS+, де FURPS - перші літери назв категорій вимог англійською мовою.

Знак плюса "+" в аббревіатурі FURPS+ охоплює додаткові категорії вимог:

- Обмеження на структуру проекту
- Вимоги до реалізації
- Вимоги до інтерфейсу
- Фізичні вимоги
- Functionality (функціональні вимоги)
- Usability (вимоги до зручності роботи)
- Reliability (вимоги до надійності)
- Performance (вимоги до продуктивності)
- Upportability (вимоги до простоти підтримки)

## **Функціональність**

Функціональні вимоги відносяться до наступних областей:

- Набір функцій
- Можливості
- Захист

## **Зручність роботи**

Вимоги до зручності роботи охоплюють такі питання:

- Людський фактор
- Естетика
- Послідовність інтерфейсу користувача
- Електронна та контекстна довідкові системи
- Майстри та агенти
- Документація

## **Надійність**

Вимоги до надійності охоплюють такі питання:

- Частота та серйозність збоїв
- Можливість відновлення
- Передбачуваність
- Точність
- Середня тривалість безперебійної роботи (MTBF)

## **Продуктивність**

Вимоги до продуктивності накладають певні умови на функціональні вимоги.

Наприклад, для певної операції можна задати такі вимоги щодо продуктивності:

- Швидкість виконання
- Ефективність
- Коефіцієнт готовності
- Точність
- Пропускна спроможність
- Час відгуку
- Час відновлення
- Використовувані ресурси

## **Зручність підтримки**

Вимоги щодо зручності підтримки охоплюють такі питання:

- Простота тестування
- Простота розширення
- Простота адаптованості
- Простота обслуговування

- Сумісність
- Простота налаштування
- Простота установки
- Простота локалізації

## Вимоги до структури проекту

Вимоги до структури проекту охоплюють питання, які стосуються структури системи, та його часто називають обмеженнями на структуру моделі.

## Вимоги до реалізації

Вимоги до реалізації відносяться до особливостей програмування та конструювання системи.

- Відповідність стандартам
- Мови реалізації
- Правила щодо цілісності баз даних
- Обмеження на ресурси
- Робочі середовища

## Вимоги до інтерфейсу

Вимоги до інтерфейсу охоплюють такі питання:

- зовнішні об'єкти, з якими має взаємодіяти система
- обмеження на формати, час очікування та інші обставини взаємодії

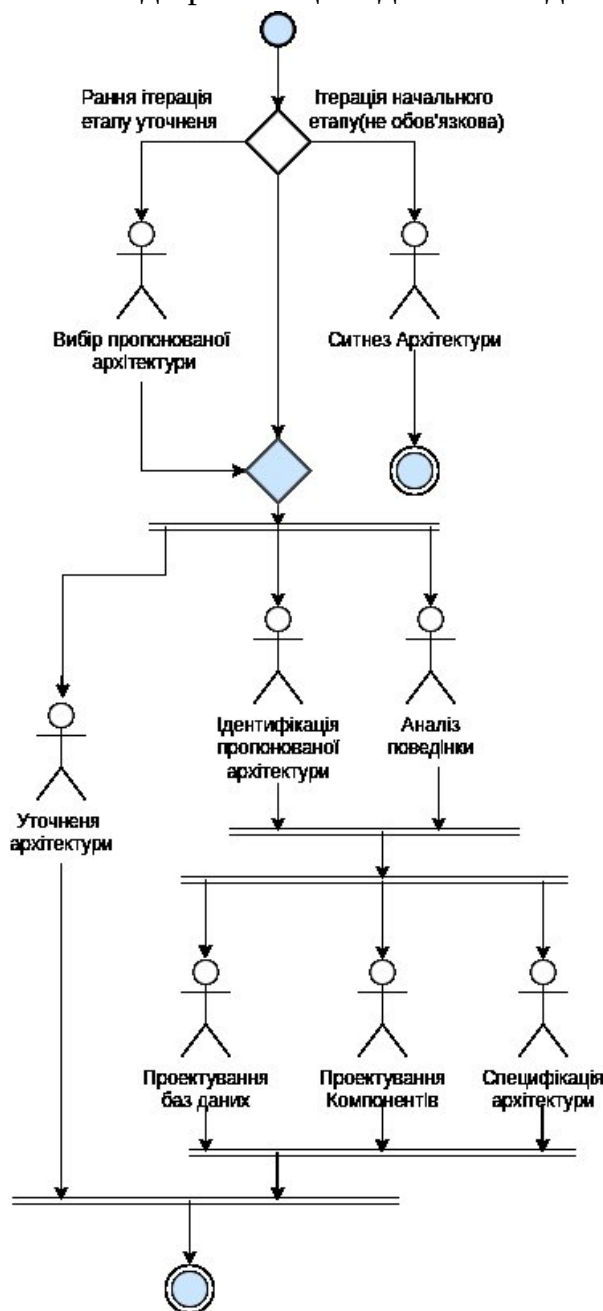
## Фізичні вимоги

Фізичні вимоги охоплюють фізичні характеристики системи:

- Кросс-платформеність
- Вага
- Критичне навантаження

Після визначення вимог проекту ми можемо перейти до **початку визначення обсягу необхідного аналізу системи**.

Я вибрав систему аналізу проекту **Classic RUP (for large projects)**(мал 1). Тому проведемо наступний опис дій у аналізі мого програмного забезпечення.



1. Графічне зображення дій при системі RUP аналізу проекту.

1. На **початковому етапі** основна мета аналізу та проектування полягає в тому, щоб зрозуміти, чи можна в принципі реалізувати систему і вибрати технології для її реалізації (Операція: **синтез архітектури**). Якщо ризики для розробки невеликі (наприклад, при розробці системи у добре відомій предметній області, оновленні існуючої системи тощо), цю процедуру виконувати не обов'язково.

Це перший етап процесу RUP, основне завдання якого полягає у відповідності цілей життєвого циклу проекту, поставлених різними зацікавленими особами.

### **Цілі**

На початковому етапі переслідуються такі основні цілі:

- Визначення змісту проекту та граничних умов, включаючи бачення, критерії приймання та розмежування того, що має входити в продукт, а що – ні.
- Відмова від найскладніших варіантів, які максимально ускладнюють.
- Розробка та демонстрація принаймні однієї потенційної архітектури для реалізації основних сценаріїв.
- Оцінка загальної вартості та тривалості реалізації проекту (включаючи детальну оцінку витрат на етапі уточнення)
- Оцінка потенційних ризиків (джерел невизначеності). розділ Концепція: ризик.
- Підготовка середовища підтримки проекту.

### **Основні операції**

На початковому етапі виконуються такі основні операції:

- Визначення змісту проекту.
  - Дана операція полягає у формалізації контексту та найважливіших вимог та обмежень у достатній мірі для розробки критеріїв приймання кінцевого продукту.
- Планування та підготовка економічного обґрунтування проекту.
  - Оцінка альтернативних варіантів управління ризиками, підбору персоналу, планування проекту та компромісів між вартістю, графіком виконання та прибутковістю проектів.
- Синтез потенційної архітектури
  - Оцінка компромісних рішень у проекті та у виборі способів реалізації окремих компонентів (купити, розробити, використовувати готовий) для прогнозування обсягу необхідних ресурсів та тривалості розробки. Мета цієї операції полягає у демонстрації можливості реалізації проекту на дослідному зразку. Роль дослідного зразка може виконувати модель, що імітує виконання необхідних функцій або початковий прототип, що дозволяє проаналізувати можливі джерела максимального ризику. Створення прототипу на початковому етапі дозволяє переконатися в тому, що проект є справді реалістичним, і що його можна буде реалізувати на етапах уточнення та побудови.
- Підготовка середовища для проекту.
  - У ході цієї операції проводиться аналіз проекту та організації, вибираються засоби розробки та приймаються рішення про оптимізацію окремих частин проекту.

Наступним етапом аналізу буде визначення **синтез архітектури**. До суттєвих для архітектури зазвичай входять питання, пов'язані з продуктивністю, масштабуванням, синхронізацією процесів та ниток, а також розподілу програмного забезпечення.

2. На початковому **етапі уточнення** основний пріоритет віддається створенню базової архітектури системи (Операція: **визначення гаданої архітектури**) , як відправної точки для повномасштабного аналізу. Якщо архітектура вже існує (наприклад, була створена в попередніх ітераціях або попередньому проекті, або імпортована з середовища розробки), акцент зміщується на уточнення існуючої архітектури (Операція: **уточнення архітектури**). Створюється початковий набір елементів, які реалізують потрібну поведінку (Операція: аналіз поведінки).

#### **Визначення гаданої архітектури:**

- Створити початковий ескіз архітектури системи
- Визначити початковий набір суттєвих елементів архітектури як основу для аналізу
- Визначити початковий набір механізмів аналізу
- Визначити початкову структуру та організацію системи
- Визначити реалізацію прецедентів для поточної ітерації
- Визначити класи аналізу за варіантами, суттєвими для архітектури
- Додати взаємодію з класами аналізу у реалізації прецедентів

#### **Операція уточнення архітектури:**

- Забезпечує природний перехід від операцій аналізу до операцій проектування та визначення наступних об'єктів:
  - Елементів проектування на основі елементів аналізу
  - Механізмів проектування на основі механізмів аналізу
- Дозволяє створити опис структури архітектури розгортання та середовища виконання системи
- Дозволяє систематизувати модель реалізації для спрощення переходу від проектування до реалізації
- Дозволяє забезпечити цілісність та одноманітність архітектури та гарантувати виконання наступних умов:
  - Нові елементи проектування, створені або виявлені в поточній ітерації, поєднуються з елементами проектування, що існували
  - Якомога раніше досягається максимальний ефект від багаторазового використання доступних компонентів та елементів проектування

3. Після ідентифікації початкових елементів починається їхнє уточнення. У процесі процедури (Операція: **проектування компонентів**) створюється набір компонентів, реалізують необхідне поведінка системи. Якщо до складу системи входить база даних, паралельно з процедурою виконується процедура (Операція: **проектування баз даних**). В результаті створюється початковий набір компонентів, які у подальшому уточнюються в (**Шаблоні можливостей: реалізація**).

### Проектування компонентів має такі цілі:

- Уточніть визначення елементів проекту деталями реалізації поведінки, що очікується від цих елементів.
- Уточніть та оновіть реалізацію прецедентів на основі новостворених елементів проекту (іншими словами, регулярно оновлюйте реалізацію прецедентів)
- Уточніть проект

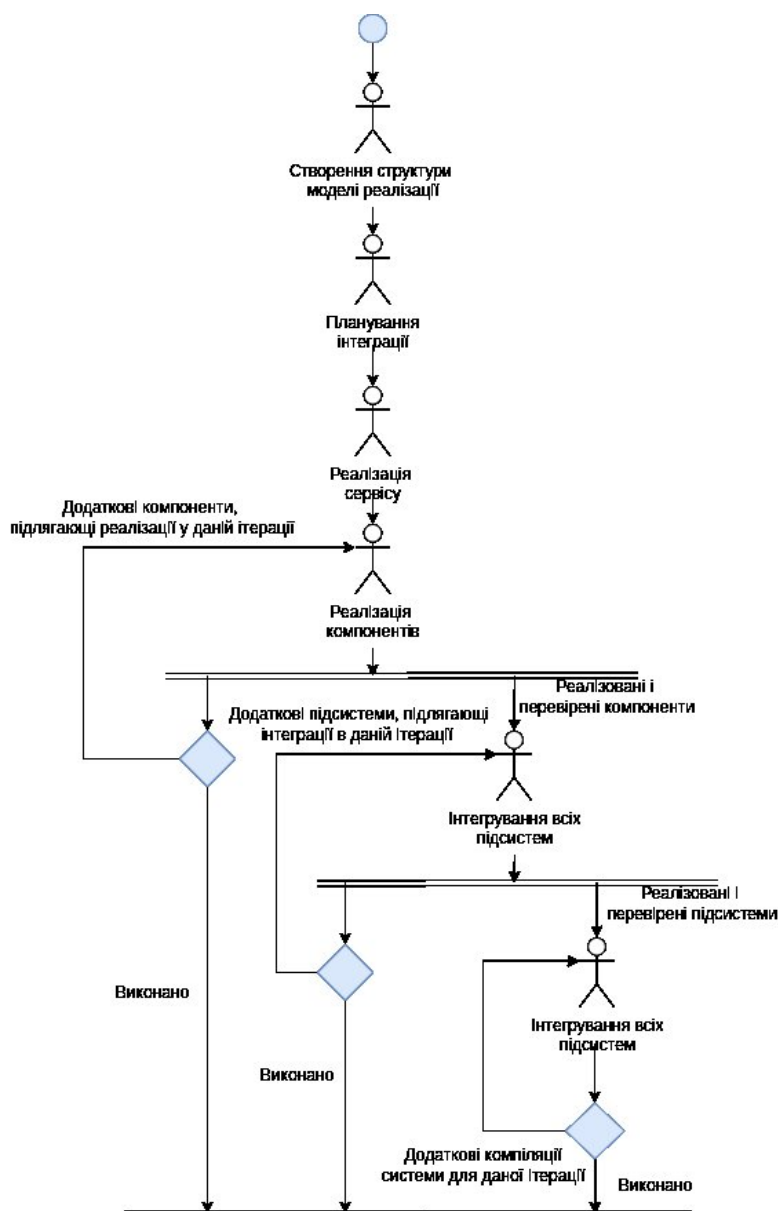
### Проектування баз даних передбачає виконання таких дій:

- Визначення постійних класів у проекті
- Проектування структур бази даних для зберігання постійних класів
- Опис механізмів та стратегій зберігання та вилучення постійних даних, що забезпечують досягнення необхідної продуктивності системи.
- База даних та механізми зберігання та вилучення постійних даних реалізуються та тестуються разом з іншими компонентами та підсистемами

додатка.

Для структурування процедур, що входять до складу **дисципліни реалізації**, сукупність операцій та продуктів роботи дисципліни називається шаблоном її можливостей.

Кожна операція призначена для вирішення деякого загального завдання, без якого неможлива ефективна реалізація. Структурування моделі реалізації виконується на початку етапу уточнення. У кожній ітерації етапу уточнення виконуватимуться такі операції: планування інтеграції, реалізація компонентів, інтеграція підсистем та інтеграція системи. Дві останні операції тісно пов'язані із тестуванням інтеграції.



2. Графічне зображення охоплює операції та потоки операцій дисципліни аналізу та проектування

## **Обґрунтування вибору типів та методик архітектурного аналізу:**

Я вибрав методологію розробки та аналізу програмного забезпечення RUP, тому що:

Ітераційна розробка програмного забезпечення в RUP передбачає поділ проекту на кілька дрібних проектів, що виконуються послідовно, і кожна ітерація розробки чітко визначена набором цілей, які мають бути досягнуті наприкінці ітерації. Кінцева ітерація передбачає, що набір цілей ітерації повинен точно співпадати з набором цілей, зазначених замовником продукту, тобто всі вимоги повинні бути виконані.

RUP досить добре формалізований, і найбільша увага приділяється початковим стадіям розробки проекту – аналізу та моделювання. Отже, ця методологія спрямовано зниження комерційних ризиків (risk mitigating) у вигляді виявлення помилок на ранніх стадіях розробки. Технічні ризики (assesses) оцінюються і «розставляються» згідно з пріоритетами на ранніх стадіях циклу розробки, а потім переглядаються з часом та з розвитком проекту протягом наступних ітерацій. Нові цілі виникають залежно від пріоритетів цих ризиків. Релізи версій розподіляються таким чином, що найпріоритетніші ризики усуваються першими.

## **Результати проведеного архітектурного аналізу:**

За вище перерахованими перевагами, ми можемо зробити висновки о результатах проведеної роботи. Визначивши цілі і характеристики проекту, ми уникнули головних ризиків під час роботи, проаналізувавши складність проекту і можливі методи реалізації поставлених задач.

Перевагою являється концентрація на виконанні вимог замовників до виконуваної програми (аналіз та побудова моделі прецедентів (варіантів використання)), що ми робили у минулій лабораторній.

Модель аналізу дозволяє очікування змін у вимогах, проектних рішеннях та реалізації у процесі розробки. Що робить проектування більш зручним і направленим на замовника.

Компонентна архітектура, що реалізується та тестується на ранніх стадіях проекту. Постійне забезпечення якості на всіх етапах розробки проекту.

## **Висновок**

В ході даної лабораторної роботи був проведений архітектурний аналіз ПЗ. Визначені основні методики та етапи аналізу, та його проведення.