МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М.Є. Жуковського

“Харківський авіаційний інститут”

Факультет програмної інженерії та бізнесу

Кафедра інженерії програмного забезпечення

**Лабораторна робота №8**

з дисципліни **«**Основи програмної інженерії**»**

на тему: «Планування проекту з допомогою діаграм Ганта»

Виконав студент 1 курсу групи № 612п

освітньої програми 121 - Інженерія програмного забезпечення

Товтин В. Б.  
Прийняв: доцент Дем’яненко О. С.

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_

Харків – 2023

**Мета роботи:** ознайомитися з функціональною методикою проектування IDEF0 на прикладі опису процесу в середовищі Draw.io. Викласти роботу для зберігання в систему контроля версій GitHub.

**Зміст звіту**

1. Постановка задачі
2. Коротке теоретичне введенння
3. Трудомісткості етапів відповідно до варіанта завдання
4. Побудована діаграма Ганта з урахуванням кількості виконавців і трудомісткості етапів
5. Короткий опис діаграм
6. Висновки

**Завдання**

1. Створити новий проект в середовищі Draw.io.
2. Побудувати діаграму Ганта для каскадної моделі життєвого циклу програмного забезпечення
3. Розмістити та зберегти роботу на GitHub. Вставити зображення в звіт.

**Теоретичні відомості**

**Каскадна модель**

Каскадна модель життєвого циклу (англ. Waterfall model) була запропонована B 1970 році Уінстоном Ройсом. Вона передбачає послідовне виконання всіх етапів проекту в строго фіксованому порядку. Перехід на наступний етап означає повне завершення робіт на попередньому етапі. Вимоги, визначені на стадії формування вимог, строго документуються у вигляді технічного завдання і фіксуються на весь час розроблення проекту. Кожна стадія завершується випуском повного комплекту документації, достатньої для того, щоб розробка могла бути продовжена іншою командою розробників.

A diagram of a company

Description automatically generated

Модель передбачає такі властивості взаємодії етапів:

Рис. 1. Каскадна модель проекту

* + модель складається з послідовно розташованих етапів; кожен етап повністю закінчується до того, як почнеться наступний; етапи не перекриваються в часі: наступний етап не починається до тих пір, поки не завершиться попередній;
  + повернення до попередніх етапів не передбачене або всіляко обмежене;
  + виправлення помилок відбувається лише на стадії тестування; - результат з'являється тільки в кінці розробки.

Критерієм появи результату є відсутність помилок та точна відповідність продукту початкової специфікації.

**Короткий опис фаз каскадної моделі.** Наведена нижче характеристика являє собою короткий опис кожної фази каскадної моделі (включаючи фази інтеграції).

**Дослідження** **концепції** – відбувається дослідження вимог на системному рівні з метою визначення можливості реалізації концепції.

**Процес системного розподілу** - може бути пропущений для систем з розробки виключно ПЗ. Для систем, в яких необхідна розробка як апаратного, так і програмного забезпечення, необхідні функції застосовуються до ПЗ і обладнання відповідно до загальної архітектури

системи.

**Процес визначення вимог** – визначаються програмні вимоги для інформаційної предметної області системи, призначення, лінії поведінки, продуктивність і інтерфейси. (В разі необхідності в процес також включено функціональний розподіл системних вимог до апаратного і програмного забезпечення.).

**Процес розроблення проекту** – розробляється і формується логічно послідовна технічна характеристика програмної системи, включаючи структури даних, архітектуру ПЗ, інтерфейсні уявлення та процесуальну (алгоритмічну) деталізацію.

**Процес реалізації** – в результаті його виконання ескізний опис ПЗ перетворюється B повноцінний програмний продукт. При цьому створюється вихідний код, база даних та документація, які лежать в основі фізичного перетворення проекту. Якщо програмний продукт являє собою придбаний пакет прикладних програм, основними діями щодо його реалізації будуть установка і тестування пакета програм. Якщо програмний продукт розробляється на замовлення, основними діями є програмування й код-тестування.

**Процес установки** – включає установку ПЗ, його перевірку й офіційне приймання замовником для операційного середовища.

**Процес експлуатації та підтримки** — запуск користувачем системи і поточне забезпечення, включаючи надання технічної допомоги, обговорення питань, що виникли з користувачем, реєстрацію запитів користувача на модернізацію і внесення змін, а також коригування або усунення помилок.

**Процес супроводження** – пов'язаний з дозволом програмних помилок, несправностей, збоїв, модернізацією та внесенням змін, що генеруються процесом підтримки. Складається з ітерацій розробки і передбачає зворотний зв'язок з надання інформації про аномалії.

**Процес виведення з експлуатації** – вивід існуючої системи з її активного використання або шляхом припинення її роботи, або завдяки її заміні новою системою або модернізованою версією існуючої системи.

**Інтегральні завдання** – включають початок роботи над проектом, моніторинг проекту та його керування, керування якістю, верифікацію і атестацію, керування конфігурацією, розроблення документації й професійну підготовку протягом усього життєвого циклу.

**Створення діаграми Ганта**

За допомогою діаграми Ганта можна скласти розклад задач проекту, а потім відстежити його хід.

У меню Файл послідовно виберіть команди Створити, Розклад, а потім - команду Діаграма Ганта. На вкладці Дата введіть кількість завдань для початку роботи, одиниці часу, які потрібно відобразити, і діапазон дат для проекту.

Параметри форматування можна змінити в будь-який момент. У меню Діаграма Ганта виберіть команду Параметри, а потім перейдіть на вкладку Формат. Замініть імена завдань за замовчуванням іменами завдань, відповідними проекту, а також замініть дати початку і закінчення завдань і тривалість.

У діаграмі Ганта в стовпці Назва завдання виділіть клітинку, яка містить завдання, яку необхідно перейменувати, а потім введіть нове ім'я. У діаграмі Ганта виділіть клітинку, яка містить дату, яку необхідно змінити, а потім введіть нову дату.

**Виконання завдання**

**Варіант 20**

3. Трудомісткості етапів:

**Формування вимог** – 10 днів

**Проектування** – 20 днів

**Реалізація** – 25 днів

**Тестування** – 4 днів

**Впровадження** – 3 днів

**Експлуатація і супровід** – 8 днів

**Всього** – 70 днів

**Кількість** **працівників** – 8 людей

4. Діаграми:

Каскадна модель процесу (рис. 4):

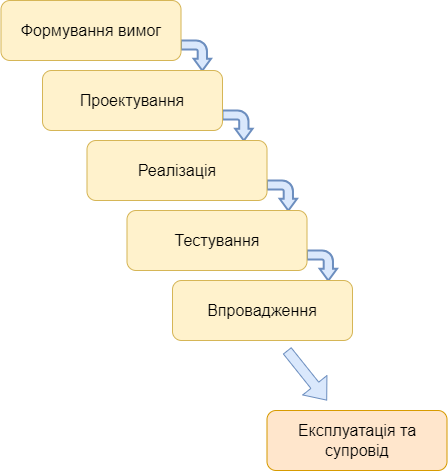


Рис. 2. Діаграма каскадної моделі

Модель відображає послідовність етапів життєвого циклу програмного забезпечення

Діаграма Ганта (рис. 3):

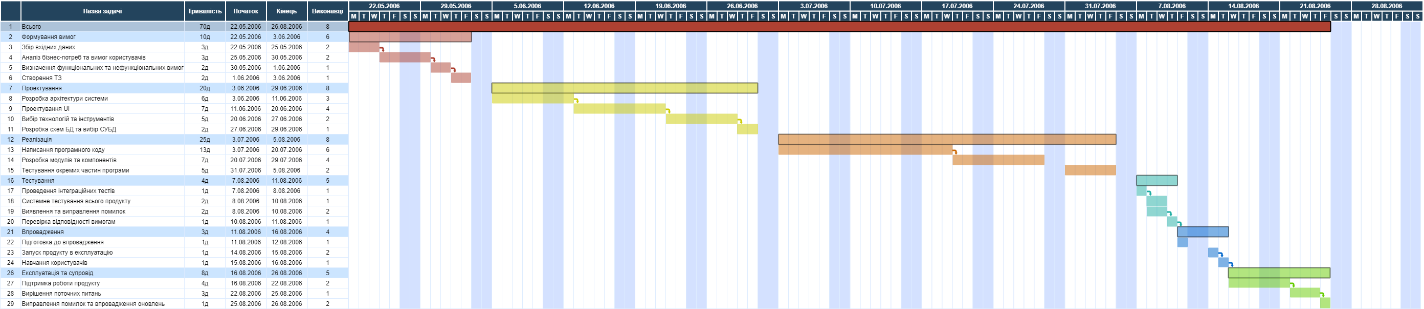


Рис. 3. Діаграма Ганта

В діаграмі відображено процеси каскадної моделі, розподілені на часовій шкалі.

Описи процесів:

Формування вимог (10 днів):

* + Збір вхідних даних – 3 дні
  + Аналіз бізнес-потреб та вимог користувачів – 3 дні
  + Визначення функціональних та нефункціональних вимог – 2 дні
  + Створення ТЗ – 2 дні

**Проектування (20 днів):**

* + Розробка архітектури системи – 6 днів
  + Проектування UI – 7 днів
  + Вибір технологій та інструментів – 5 днів
  + Розробка схем БД та вибір СУБД – 2 дні

**Реалізація (25 днів):**

* + Написання програмного коду – 13 днів
  + Розробка модулів та компонентів – 7 днів
  + Тестування окремих частин програми – 5 днів

**Тестування (4 дні):**

* + Проведення інтеграційних тестів – 1 день
  + Системне тестування всього продукту – 2 дні
  + Виявлення та виправлення помилок – 2 дні
  + Перевірка відповідності вимогам – 1 день

**Впровадження (3 дні):**

* + Підготовка до впровадження – 1 день
  + Запуск проекту в експлуатацію – 1 день
  + Навчання користувачів – 1 день

**Експлуатація та супровід (8 днів):**

* + Підтримка роботи продукту – 4 дні
  + Вирішення поточних питань – 3 дні
  + Виправлення помилок та впровадження оновлень – 1 день

Діаграми було завантажено на GitHub - https://github.com/vadim-ghostman/bse/tree/main/lab8

**Висновок**

Під час виконання даної лабораторної роботи я ознайомився з діаграмою Ганта, виконавши розбір каскадної моделі життєвого циклу програмного забезпечення і розподіливши всі задачі і процеси по дням і виконавцям