

ДОДАТОК А
Графічний матеріал

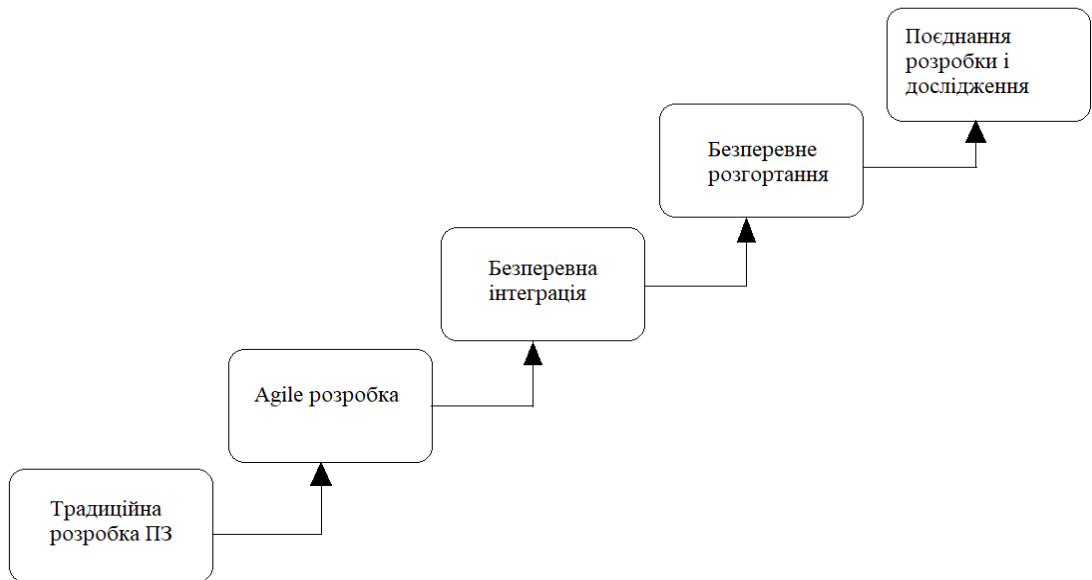


Рисунок А.1 – Етапи впровадження безперервної доставки

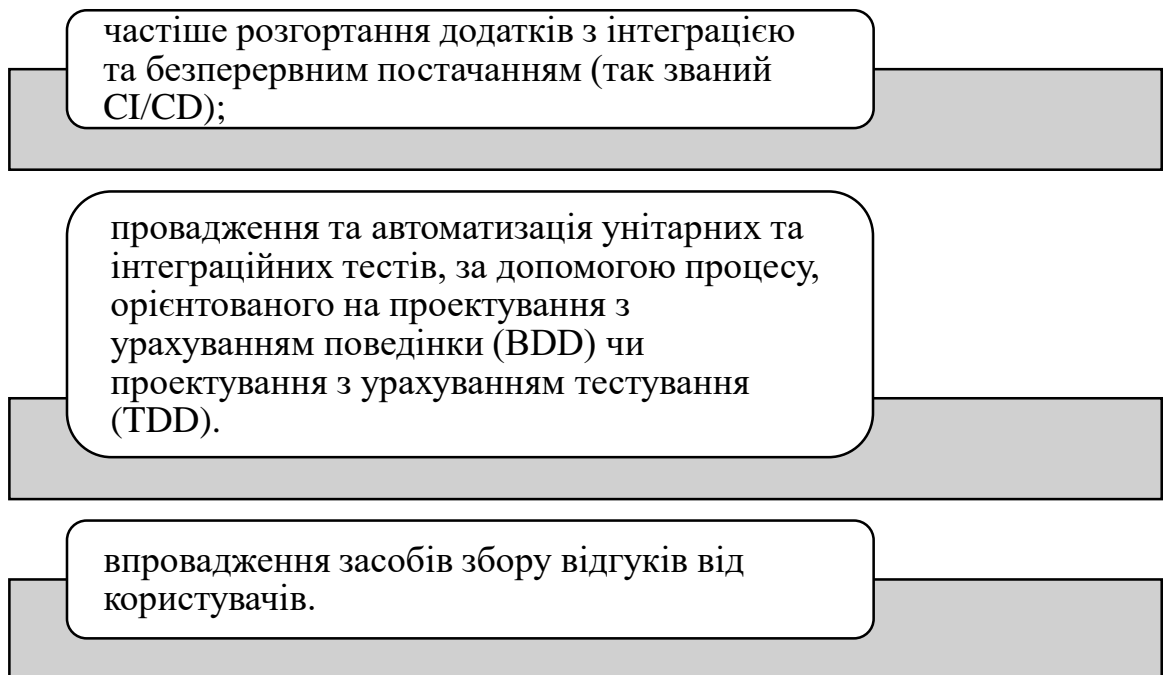


Рисунок А.2 – Елементи для покращення зв'язку між Dev та Ops



Рисунок А.3 – Співпраця між Dev та Ops, процеси та використання інструментів.

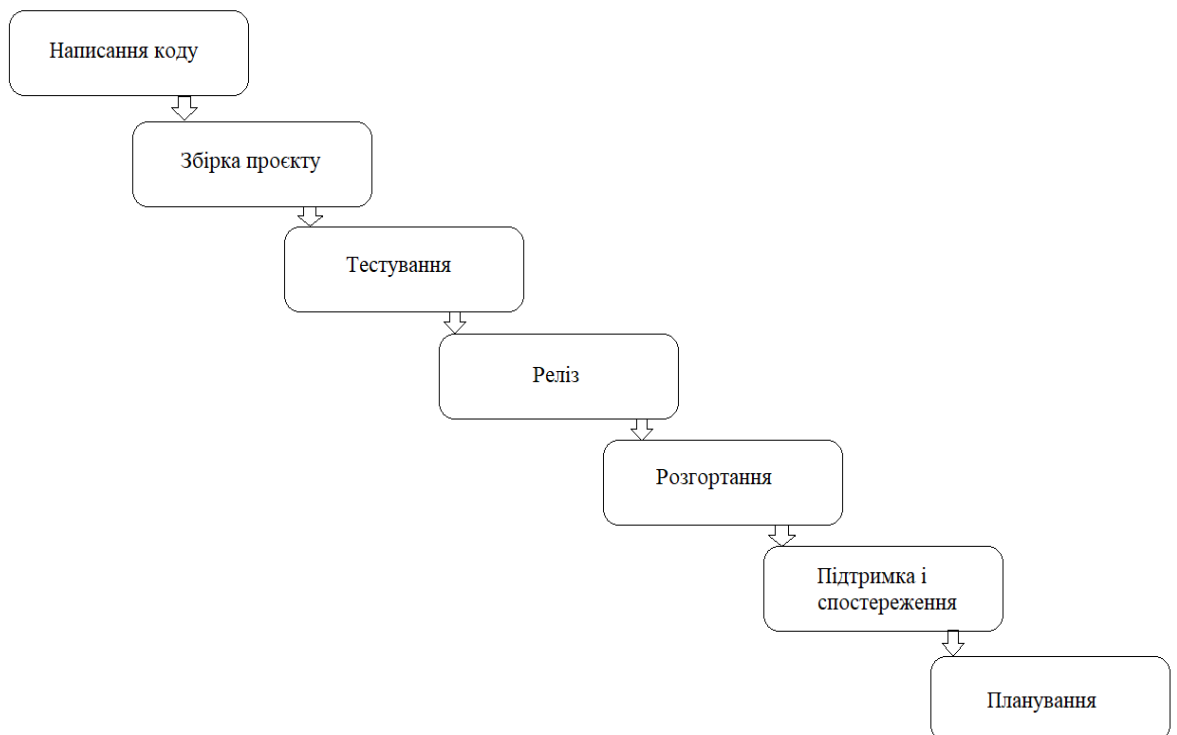


Рисунок А.4 – Етапи безперервної інтеграції

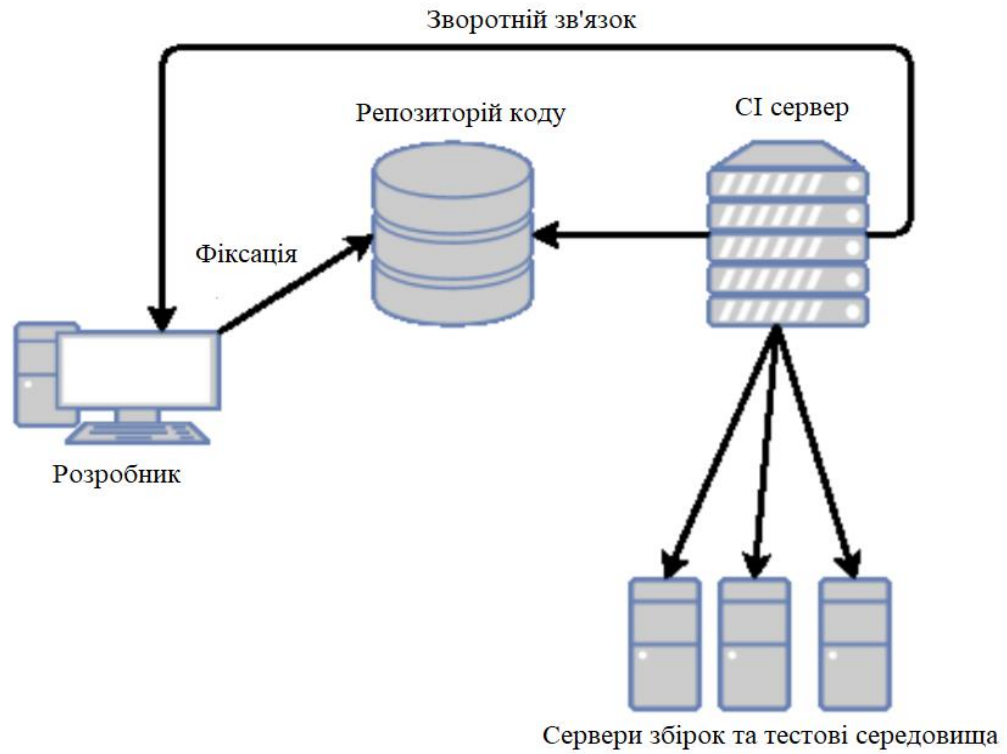


Рисунок А.5 – Середовище СІ

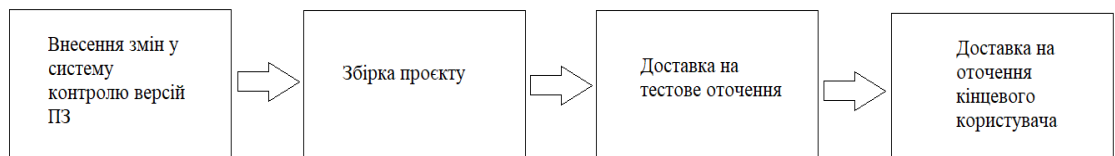


Рисунок А.6 – Етапи безперервного постачання

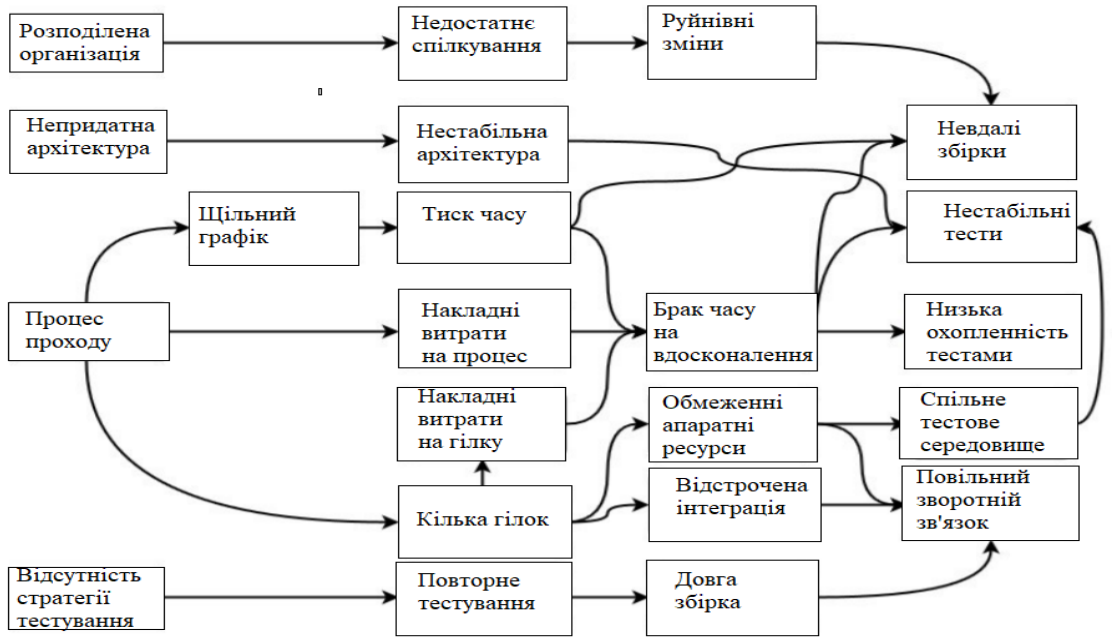


Рисунок А.7 – Огляд прямих ознак, основних причин та механізму між ними

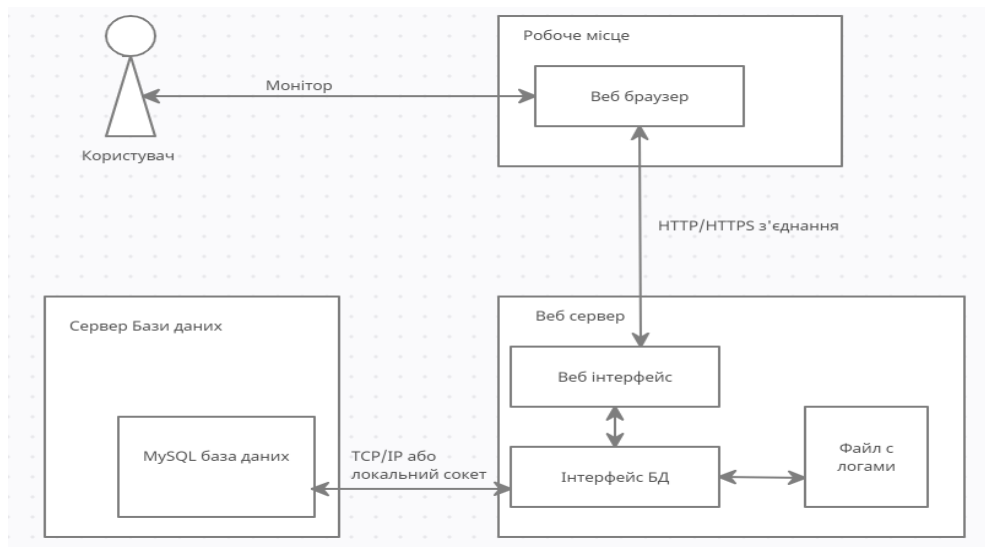


Рисунок А.8 – Процес розгортання оточення

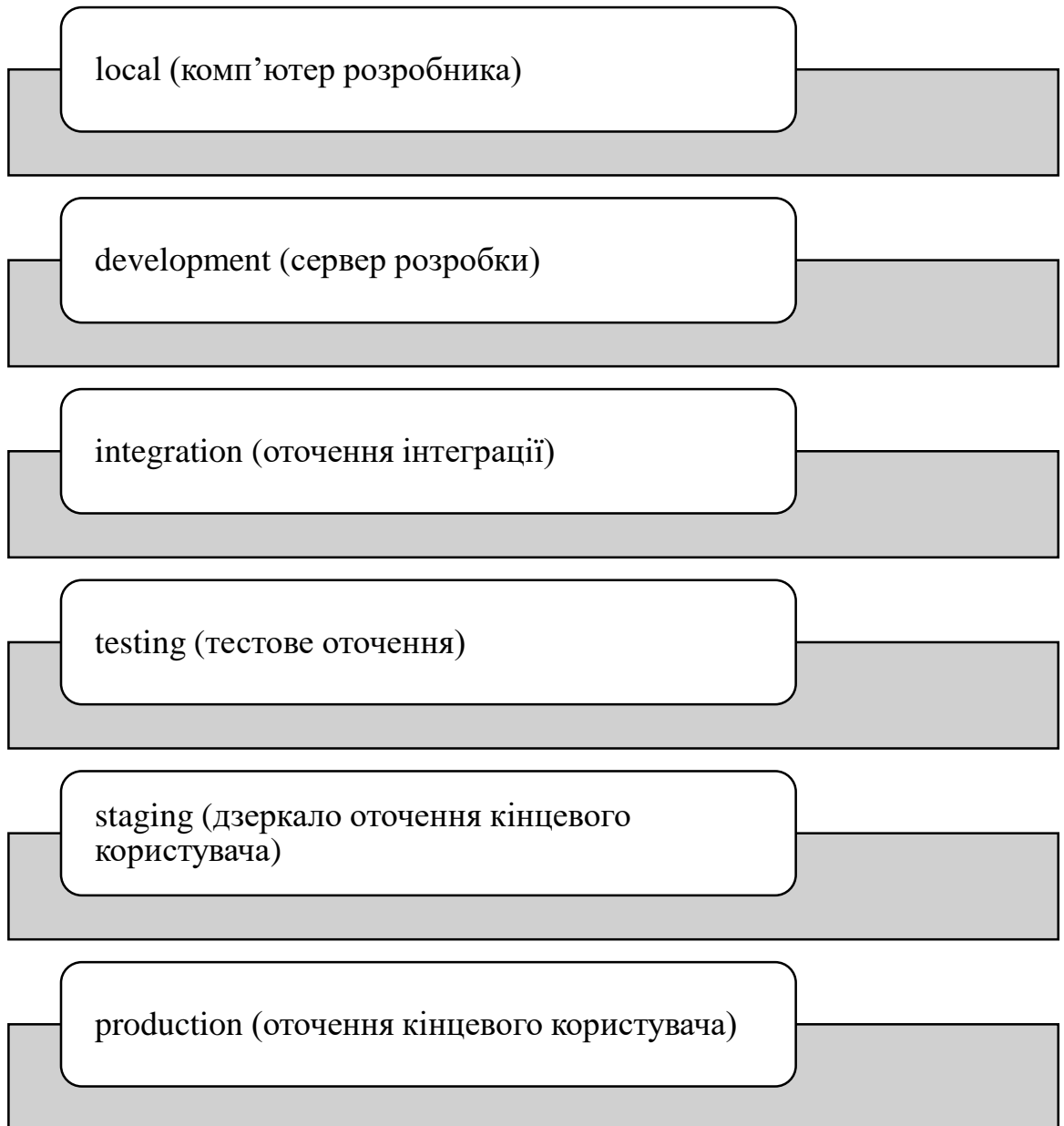


Рисунок А.9 – Види оточень

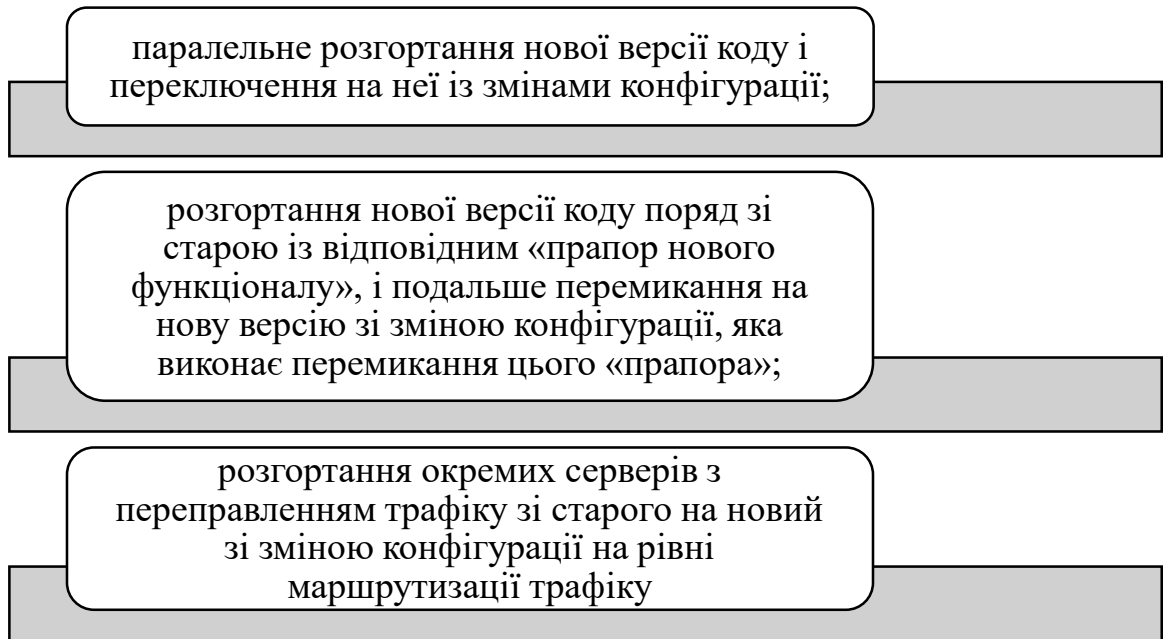


Рисунок А.10 – Способи розгортання ПЗ на оточенні кінцевого користувача

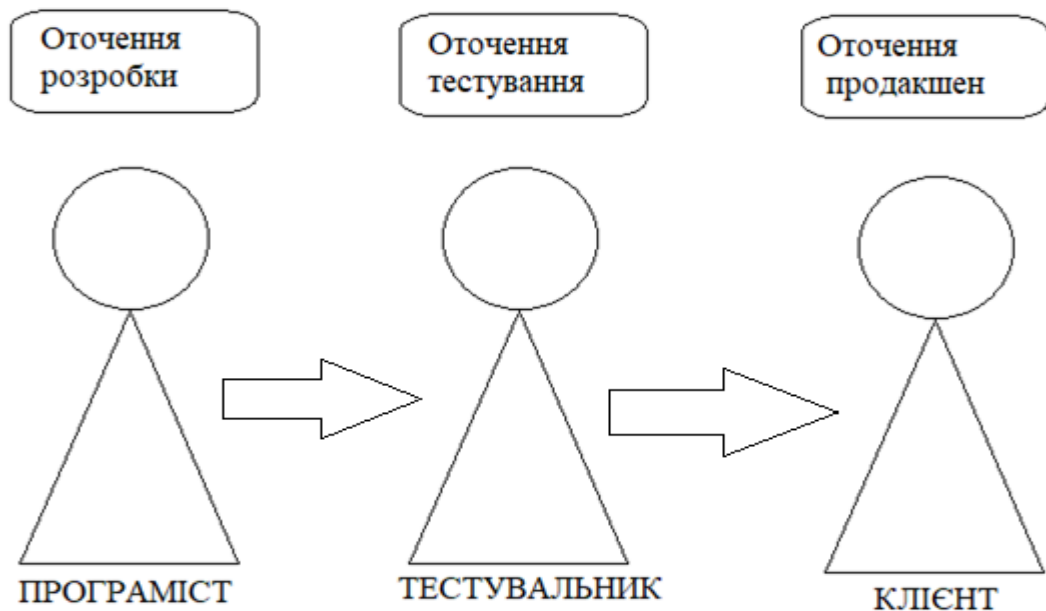


Рисунок А.11 – Схематичне зображення поділу на оточення 3-х ярусної архітектури

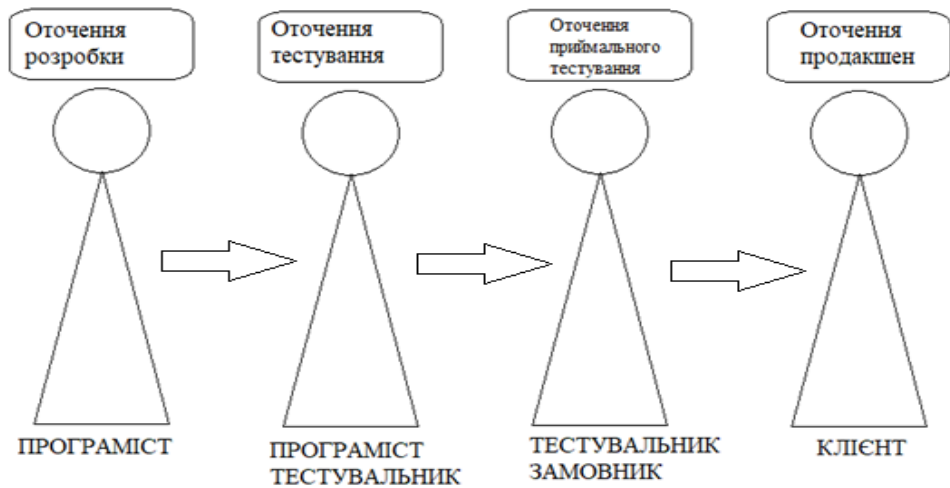


Рисунок А.12 – Схематичне зображення DTAP архітектури

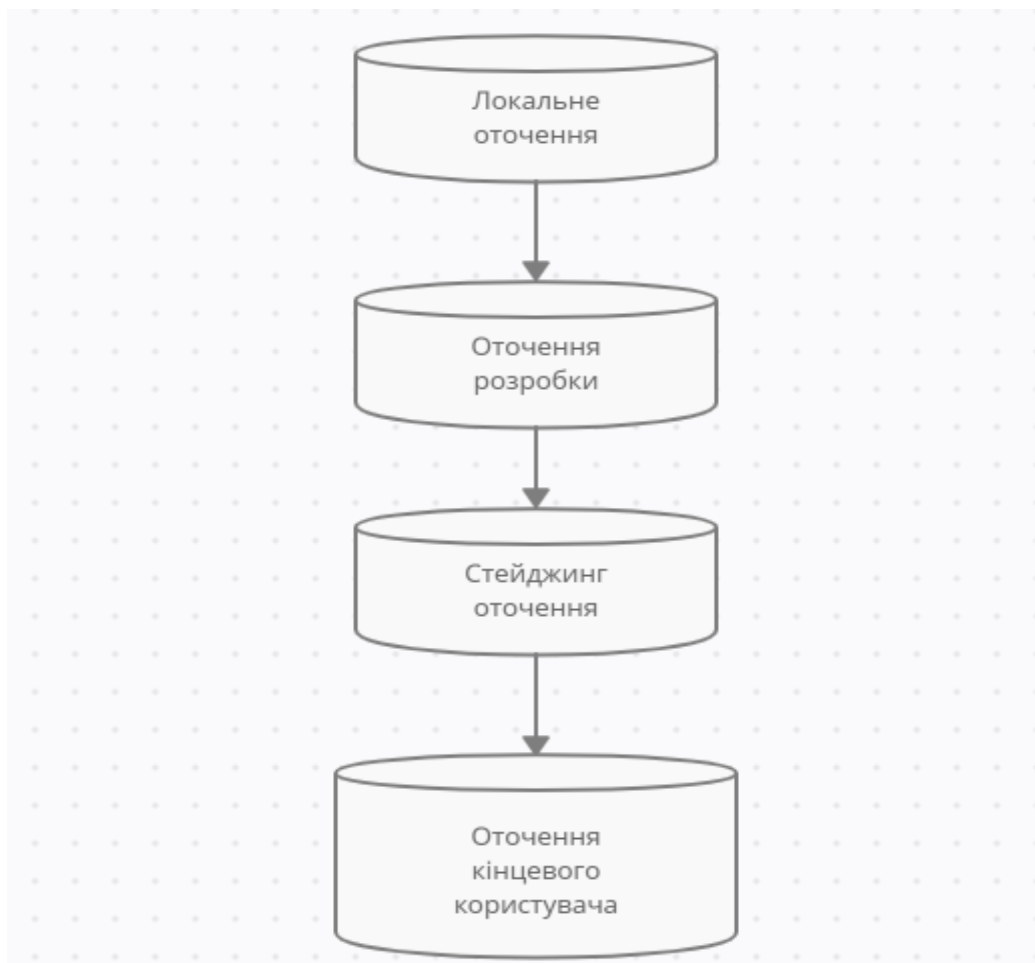


Рисунок А.13 – Архітектура оточень розгортання до оптимізації

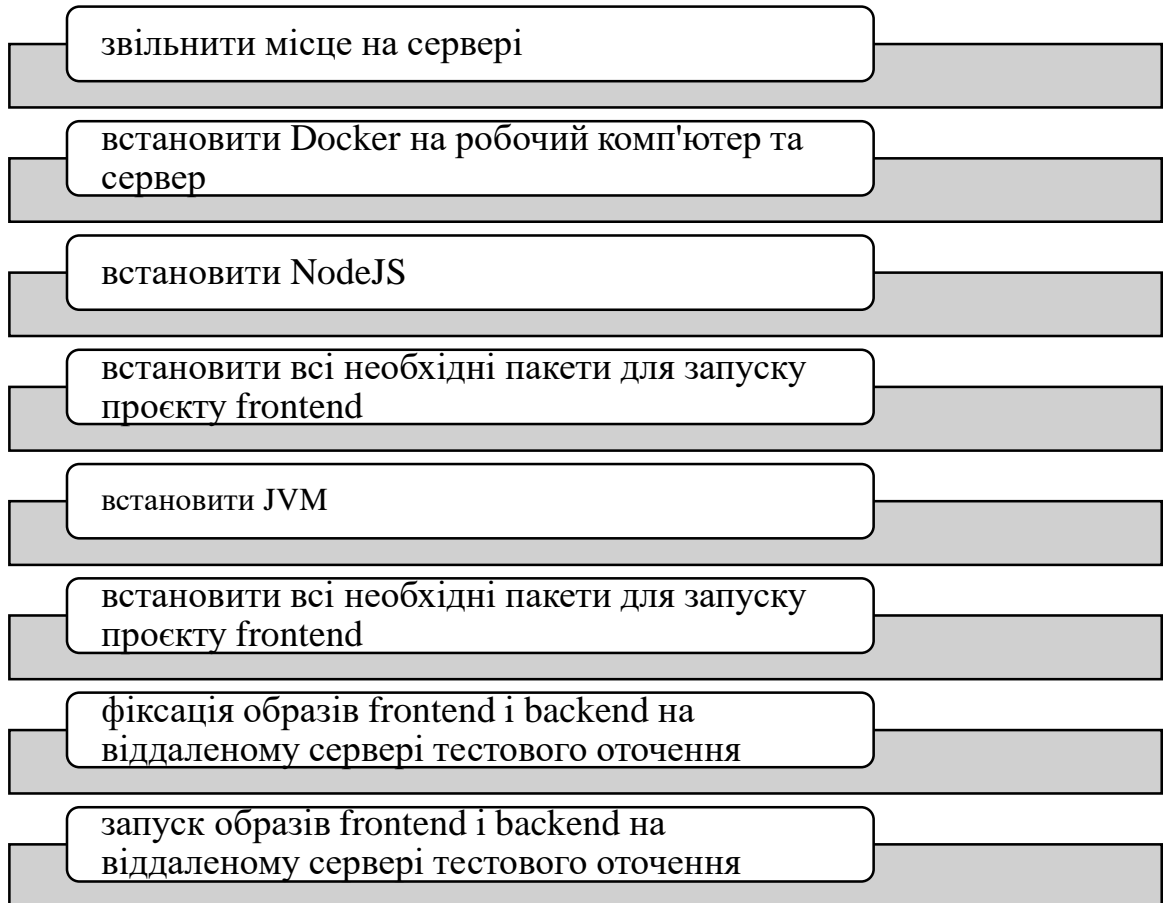


Рисунок А.14 - Етапи додавання тестового оточення та розгортання на ньому проєкту

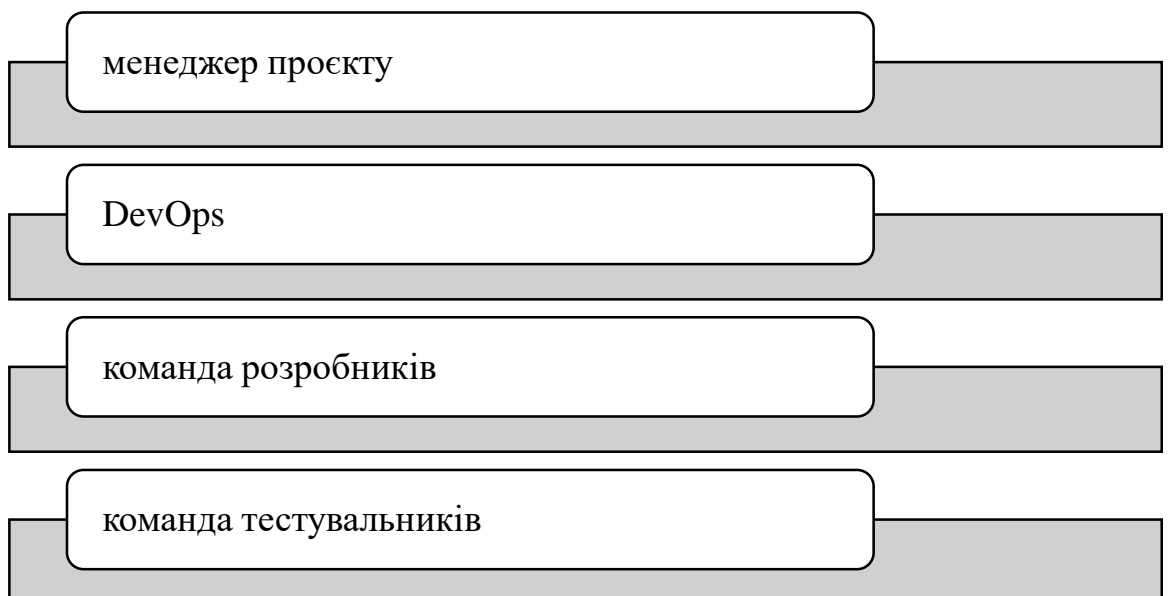


Рисунок А.15 – Члени команди по додаванню нового оточення

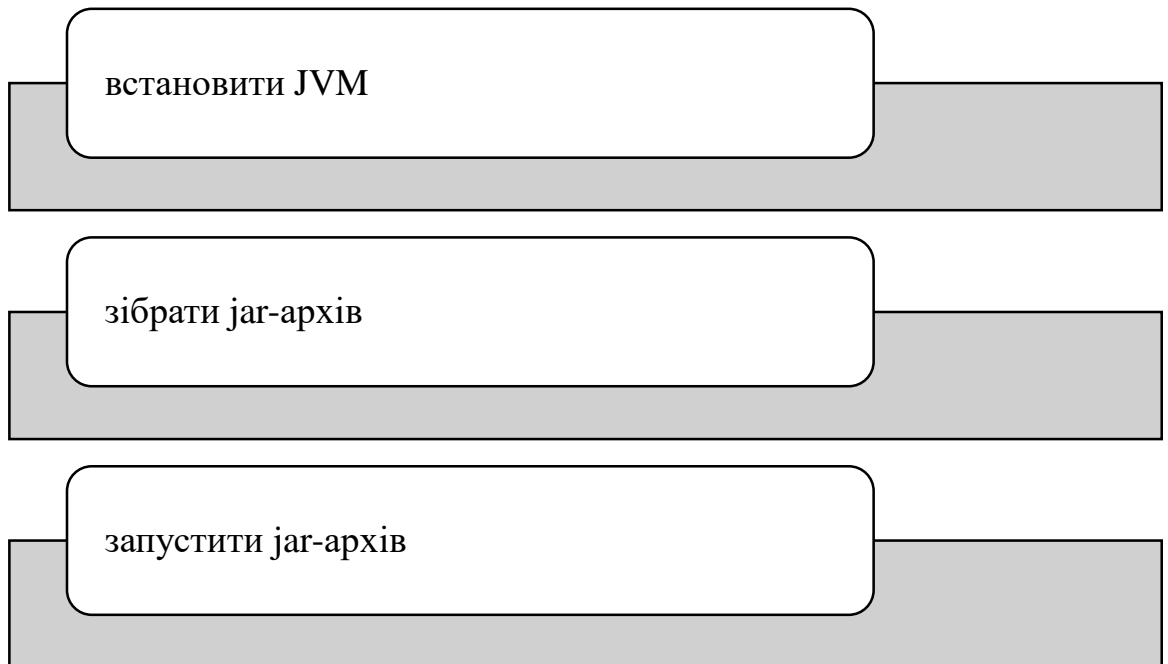


Рисунок А.16 - Послідовність дій при створенні образу backend

Етап проекту	Початок	Тривалість	Затримка	Кінець
Постановка задачі	01.09.2021	1	0	01.09.2021
Написання ТЗ	02.09.2021	11	0	12.09.2021
Розгортання додаткового оточення	13.09.2021	3	-1	16.09.2021
Налаштування оточення	17.09.2021	5	0	22.09.2021
Тестування	23.09.2021	4	0	27.09.2021
Планування задач	28.09.2021	1	0	28.09.2021
Розробка нового функціоналу	29.09.2021	15	3	15.10.2021
Тестування нового функціоналу на новому оточенні	16.10.2021	2	-1	18.10.2021
Проведення регресійного тестування на оточенні staging	19.10.2021	5	-2	24.10.2021
Доставка нового функціоналу на оточення кінцевого користувача	25.10.2021	1	0	25.10.2021
Поверхнєве тестування на оточенні кінцевих користувачів	25.10.2021	1	0	25.10.2021
Складання звіту	26.10.2021	1	0	26.10.2021

Рисунок А.17 - Декомпозиція робіт проекту (WBS)

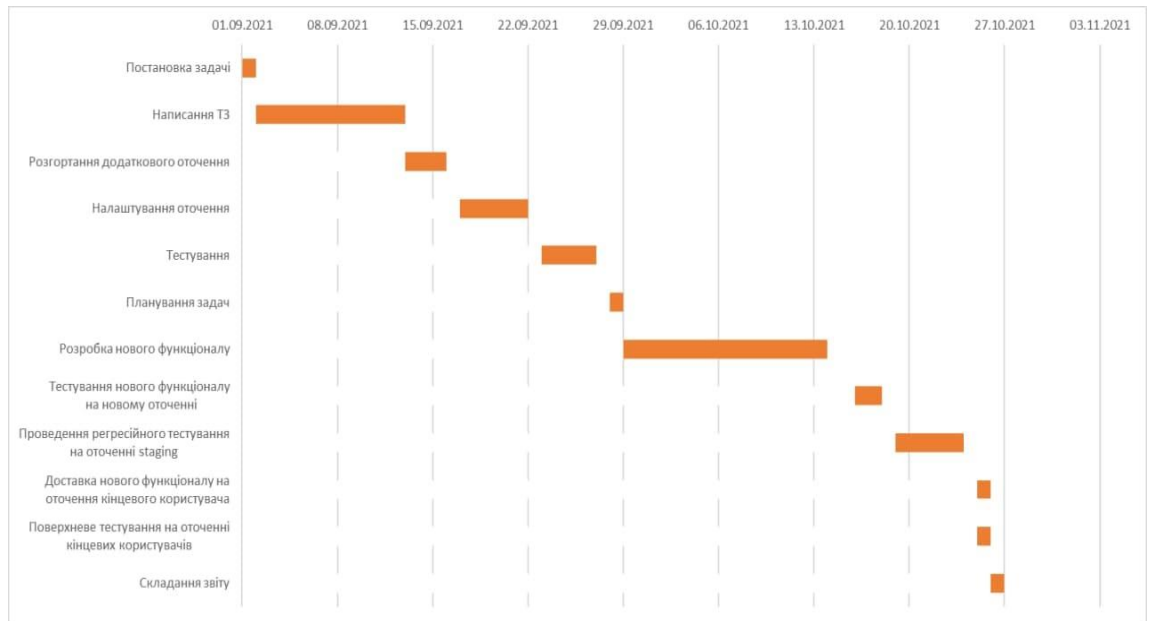


Рисунок А.18 – Діаграма Ганта із критичним шляхом



Рисунок А.19 – Організаційна структура проєкту (OBS)

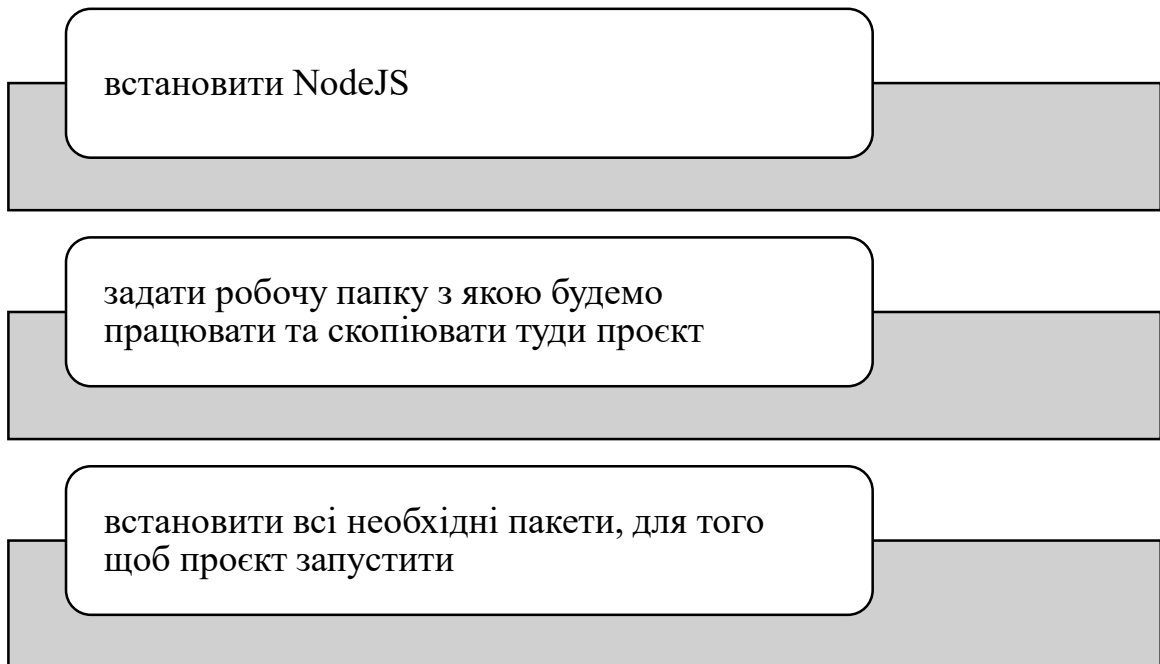


Рисунок А.20 – Послідовність дій при створенні образу frontend

```

1 FROM node:alpine
2 WORKDIR /usr/app/front
3 EXPOSE 3000
4 COPY ./ ./
5 RUN npm install
6 CMD ["npm", "start"]

```

Рисунок А.21 – Встановлення необхідних пакетів та запуск частини frontend проєкту

```

3
4 docker build -t rebounder-chain-frontend .
5

```

Рисунок А.22 – Збірка frontend

```

1 Step 7/7 : CMD ["npm", "start"]
2 ----> Running in ee0e8a9066dc
3 Removing intermediate container ee0e8a9066dc
4 ----> b208c4184766
5 Successfully built b208c4184766
6 Successfully tagged rebounder-chain-frontend:latest

```

Рисунок А.23 – Результат збірки

```
4 docker run -p 8080:3000 rebounder-chain-frontend
```

Рисунок А.24 – Запуск образу frontend

```
1 Auz-Yuliana:rebounder-chain-frontend xpendence$ docker run -p 8080:3000 rebounder-chain-frontend
2
3 > rebounder-chain-frontend@0.1.0 start /usr/app/front
4 > react-scripts start
5
6 Starting the development server...
7
8 Compiled successfully!
9
10 You can now view rebounder-chain-frontend in the browser.
11
12 Local:          http://localhost:3000/
13 On Your Network: http://172.17.0.2:3000/
14
15 Note that the development build is not optimized.
16 To create a production build, use npm run build.
```

Рисунок А.25 – Результат виконання запуску образу

```
1 FROM openjdk:8-jdk-alpine
2 LABEL maintainer="2262288@gmail.com"
3 VOLUME /tmp
4 EXPOSE 8099
5 ARG JAR_FILE=build/libs/rebounder-chain-backend-0.0.2.jar
6 ADD ${JAR_FILE} rebounder-chain-backend.jar
7 ENTRYPOINT ["java","-jar","/rebounder-chain-backend.jar"]
8
```

Рисунок А.26 – Встановлення необхідних пакетів частини backend проекту

```
1 docker build -t rebounder-chain-backend .
2 docker run -p 8099:8099 rebounder-chain-backend
```

Рисунок А.27 – Збірка та запуск backend частини проекту

```
1 docker tag rebounder-chain-backend xpendence/rebounder-chain-backend:0.0.2
2 docker tag rebounder-chain-frontend xpendence/rebounder-chain-frontend:0.0.1
```

Рисунок А.28 – Команда присвоєння тега образам

```

4 docker push xpendence/rebounder-chain-backend:0.0.2
5 docker push xpendence/rebounder-chain-frontend:0.0.1
6

```

Рисунок А.29 – Команда фіксації образів у Docker

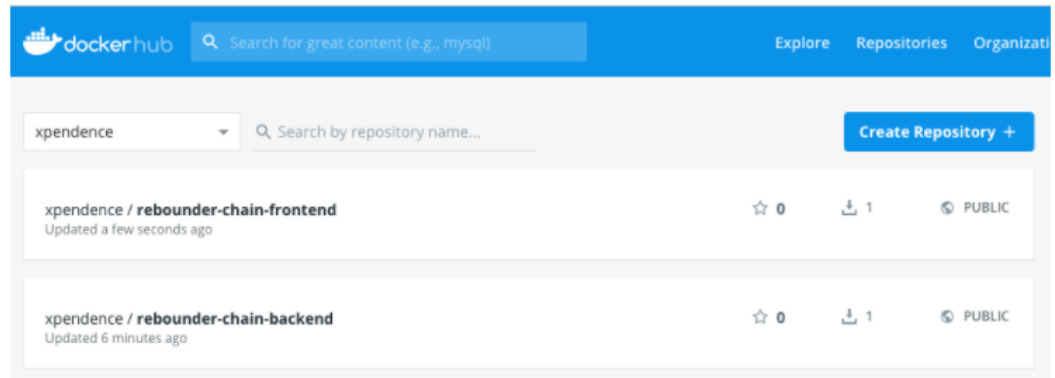


Рисунок А.30 – Відображення доданих образів у Docker

```

1 docker run -d -p 8099:8099 xpendence/rebounder-chain-backend:0.0.2
2 docker run -d -p 8080:3000 xpendence/rebounder-chain-frontend:0.0.1

```

Рисунок А.31 – Запуск образів на віддаленому сервері

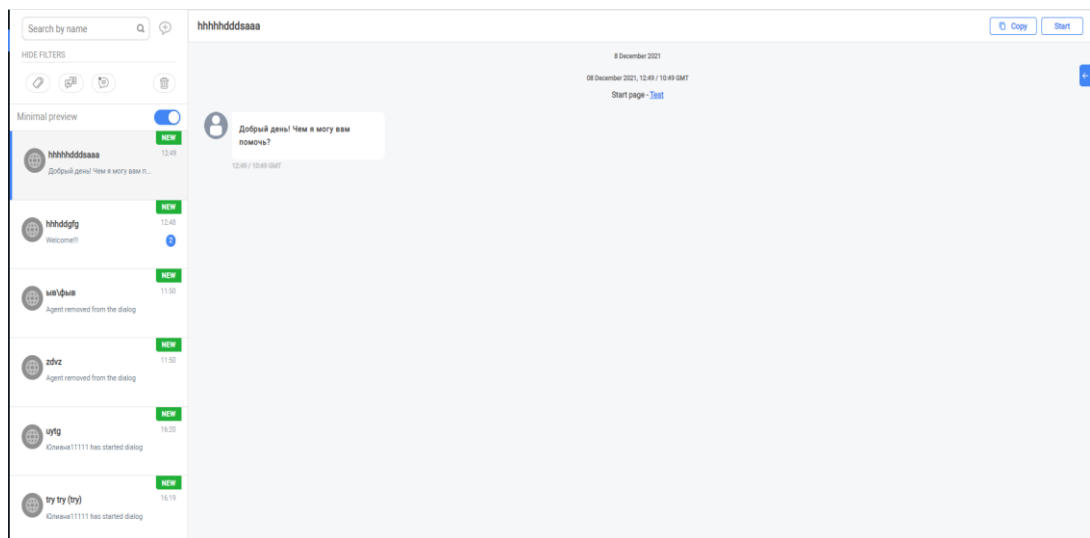


Рисунок А.32 – Успішне розгортання проєкту на тестову оточенні

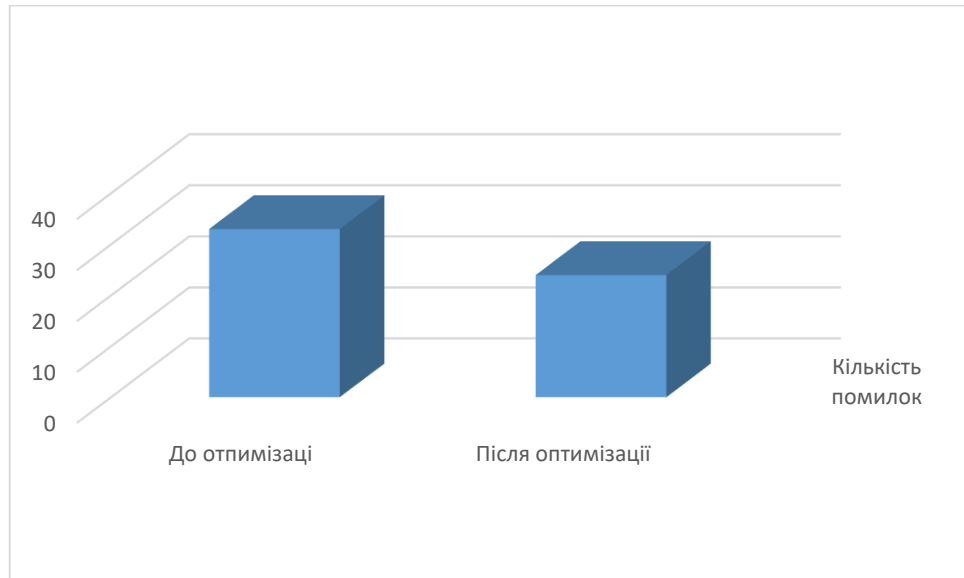


Рисунок А.33 – Динаміка зміни кількості помилок

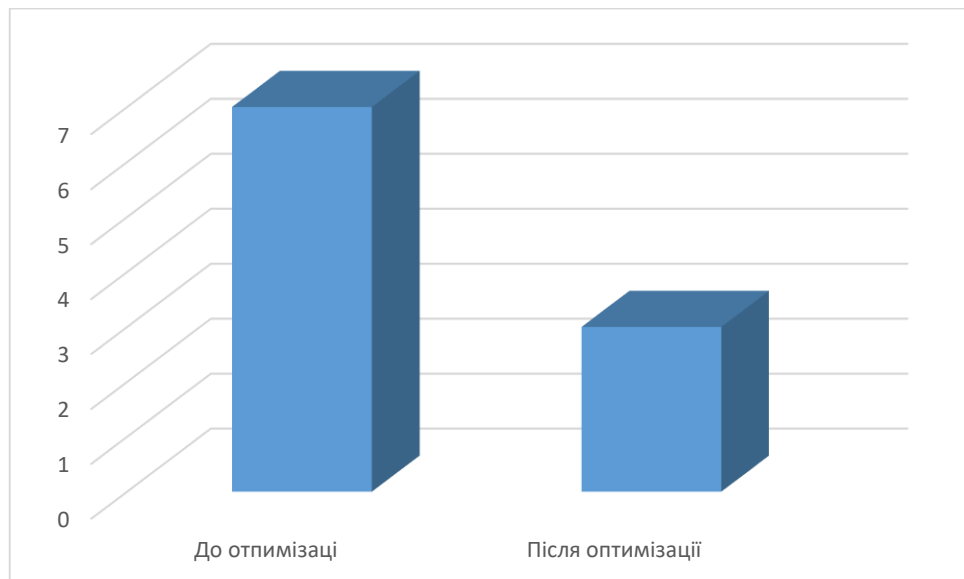


Рисунок А.34 – Динаміка зміни затримки випуску нового функціоналу до користувача

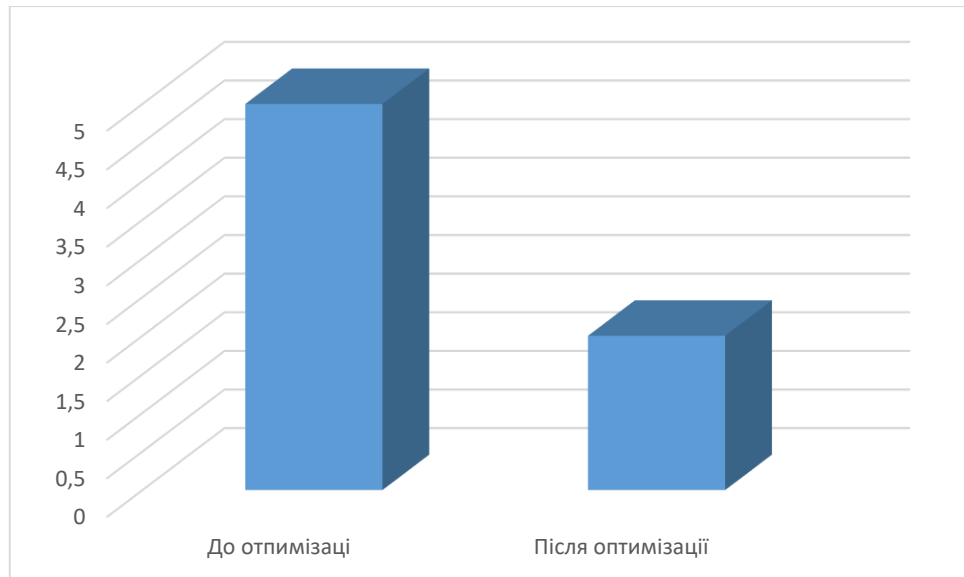


Рисунок А.35 – Динаміка зміни часу тестування нового функціоналу

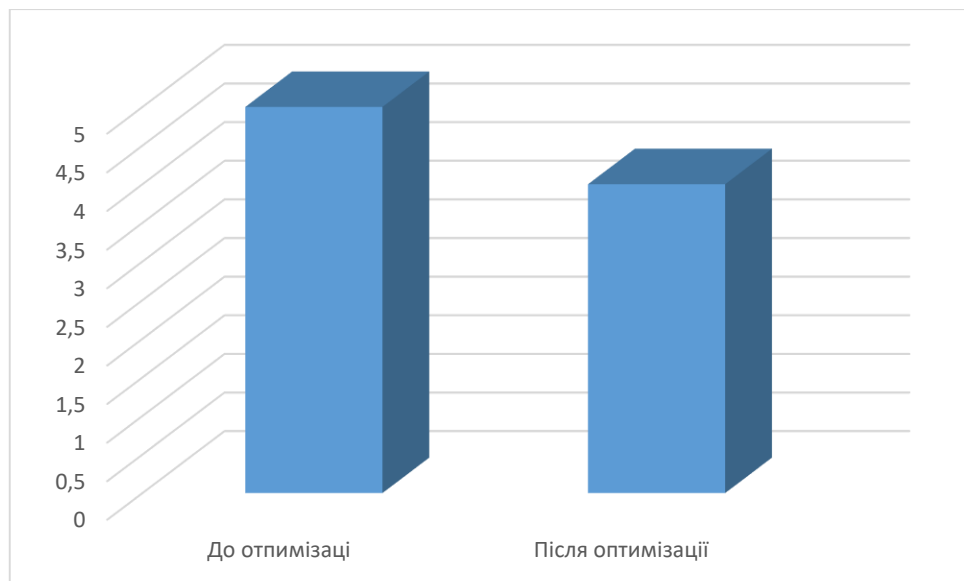


Рисунок А.36 – Динаміка зміни часу регресійного тестування

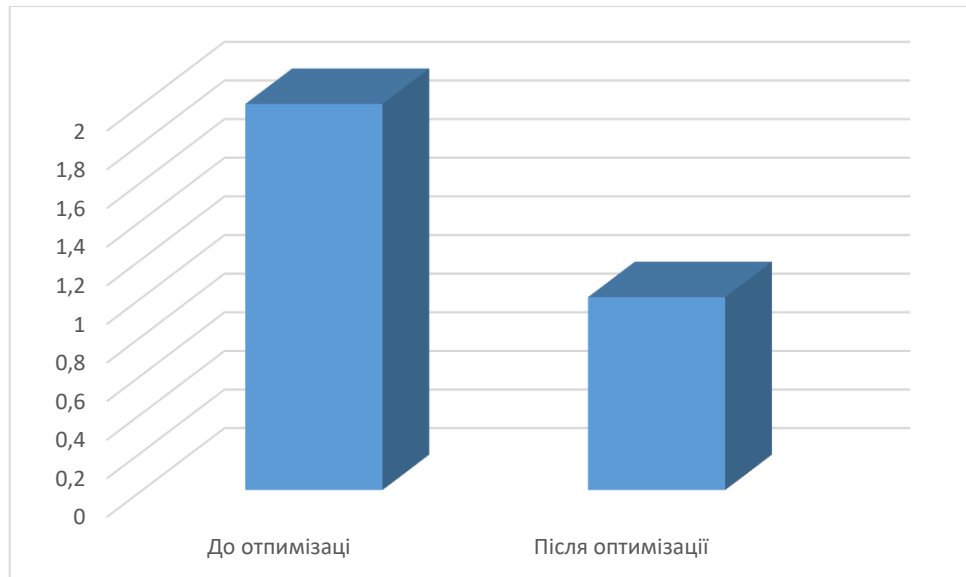


Рисунок А.37 – Динаміка зміни кількості помилок, які потрапили на оточення користувачів

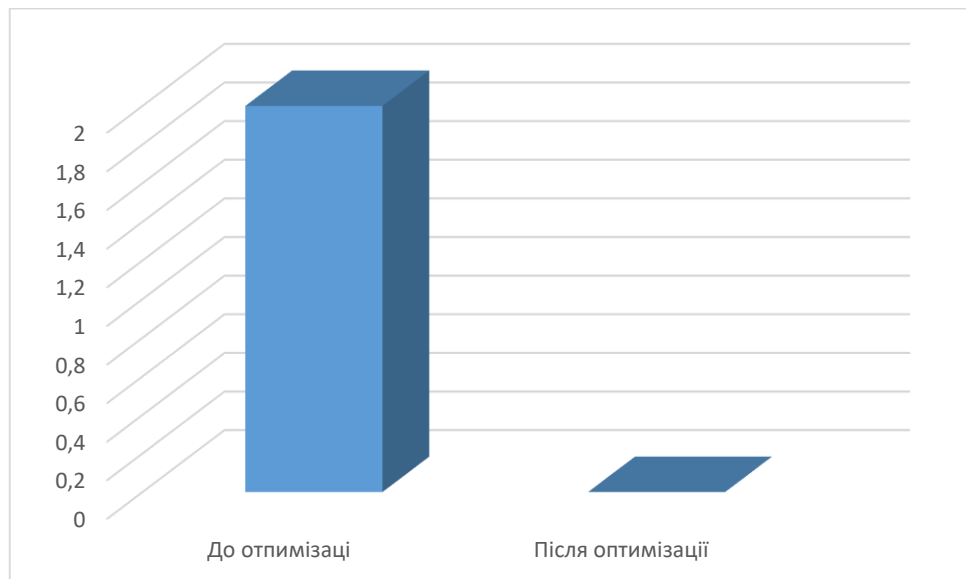


Рисунок А.38 – Динаміка зміни кількості додаткових релізів