

Міністерство освіти і науки, молоді та спорту України

Національний Технічний Університет України

«Київський Політехнічний Інститут»

Навчально-науковий комплекс

«Інститут прикладного системного аналізу»

Кафедра системного проектування

«**ТЕХНОЛОГІЇ ЗАХИСТУ ІНФОРМАЦІЇ»**

Лабораторна робота № 3  
Розробка життєвого циклу проекту та системи багтрекінгу  
Розробка поетапного плану проекту.

Виконав:  
студент 4 курсу,

Група ДА-61

Воловик Іван

м. Київ

2019

**Частина 1. Розробка життєвого циклу проекту та системи багтрекінгу.**

Мета роботи: Вивчити типові життєві цикли розробки програмного забезпечення. Усвідомити методику роботи з багтрекінгом, навчитися створювати питання, задавати їх статус, а також навчитися оперувати статистикою багтрекінга.

Задача: Вивчити зміст прикладу інформаційної системи та створити опис життєвого циклу розробки відповідного програмного забезпечення. Використовуючи систему багтрекінга MantisBT або подібну необхідно ознайомитися веденням журналу для повного циклу розробки та тестування, відтворити статуси для питання (ticket) та ролі у процесі розробки програмного забезпечення.

**Частина 2. Розробка поетапного плану проекту.**

Мета роботи: Скласти і описати поетапний план проекту.

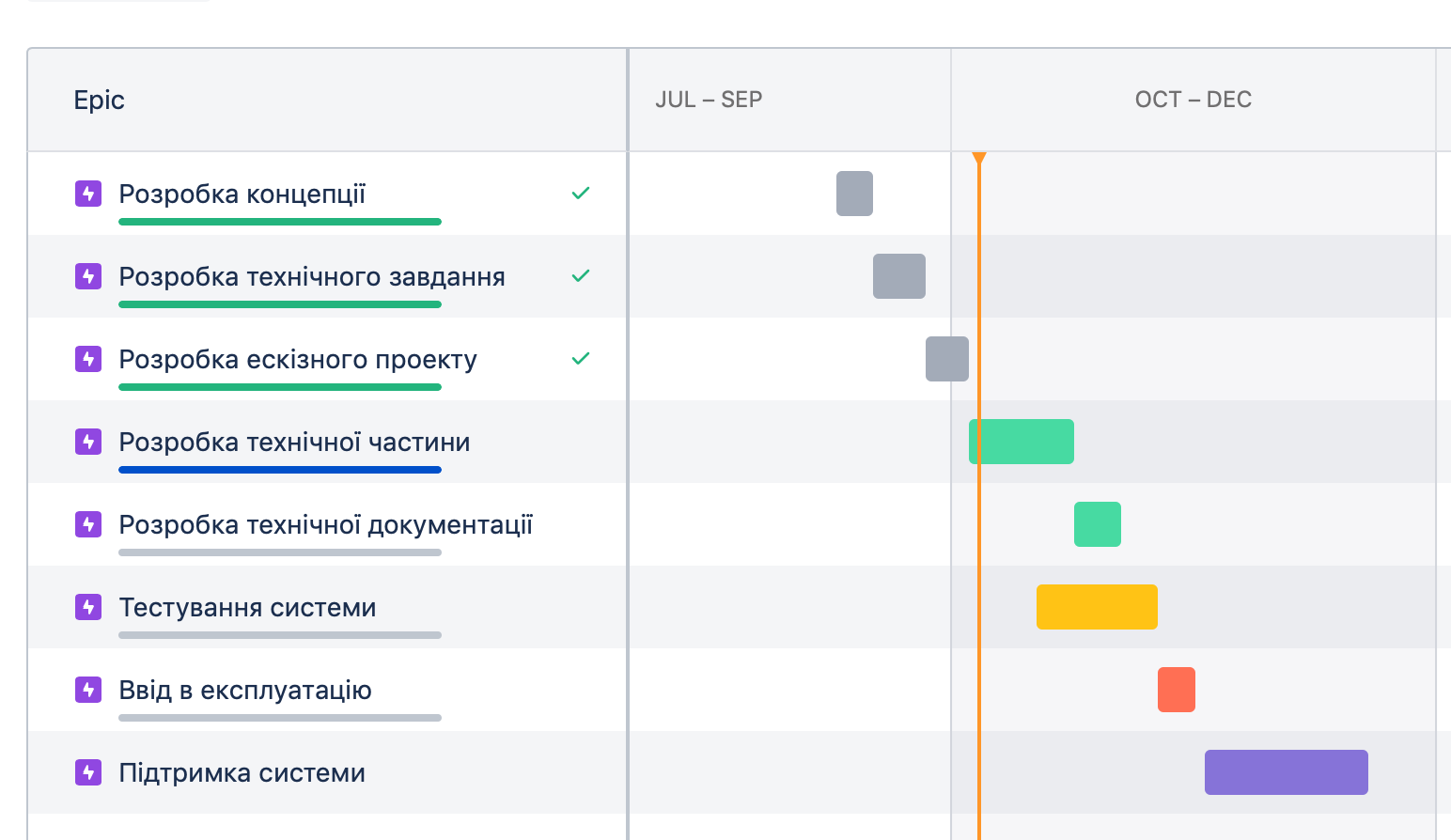
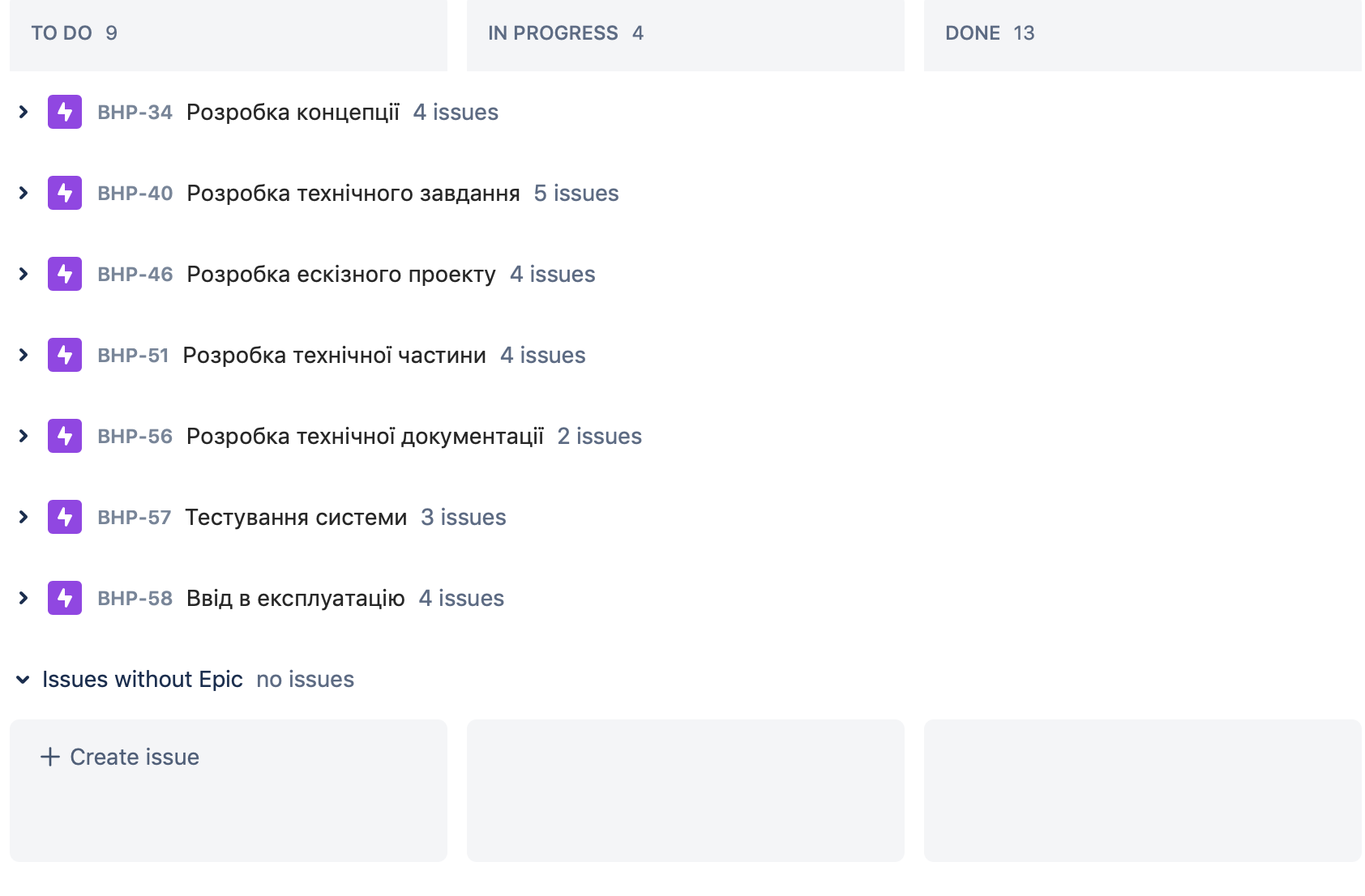
Задача: розробити поетапний план проекту, створити діаграму Ганта, розбити проект на етапи, визначити обсяги постачання готового продукту в кожному проекті, розробити відповідну документацію. Для кожного етапу розрахувати ризики та розробити список анти ризикових заходів.

**Реалізація**   
**Частина 1.** Життєвий цикл для проекту, використовуючи Баг-трекінгова система - Jira Software. A screenshot of a cell phone

Description automatically generated

За кожен етап розробки проекту відповідають назначені працівники (в нашому варіанті такі): Менеджер проекту, Програміст, Тестувальник (QA), Дизайнер інтерфейсів.

|  |  |  |  |
| --- | --- | --- | --- |
| **Project charter** | | | |
| Project name | Customer Loyalty App (CLA) | | |
| Project description | Додаток надає можливості продавцю помічати у додатку користувача придбані ним товари, і надавати покупцю знижки та бонуси. Програма зорієнтована на некомерційне використання користувачами всіх вікових категорій̆ для отримання власної електронної бонусної програми. | | |
| Project Manager | Воловик І.М. | Data Approved | 2019/10/10 |
| Project Sponsor | SoupCulture | Signature | Volovyk |
| Business case | | Expected Goals /Deliverables | |
| Реалізація додатку на платформі IOS 100, мова програмування – Swift 5, база даних – MongoDb, серверна частина – Node Js.  Додаток буде містить лише п’ять активних інтерфейсів з мінімальним дизайном. Реалізується на основі мінімального бюджету та не потребує великої кількості людей на проекті.  Більше інформації в передпроектної̈ документацій̈. | | * Мінімальний, але функціональний дизайн, який задовольнить користувача. * Зрозумілість використання. * Швидкість доступу до вмісту. * Швидка та дешева реалізація додатку. * Збереження даних на сервері. | |
| Team members | |
| Name | Role |
| B-Fly | Програміст |
| Snowball | Тестувальник |
| Flower | Дизайнер |
| Risk and Constraints | | Milestones | |
| * потреба більшого бюджету (поламка серверу, інфляція, інше). * Нехватка кадрів/досвіду. * Незацікавленість користувачів. * Технічні несправності. * Нехватка часу. | | * Розробка технічного завдання. * Розробка ескізного проекту. * Розробка технічної частини. * Розробка технічної документації. * Тестування системи. * Ввід в експлуатацію. * Підтримка системи. | |

Roadmap для проекту CLP, описує основі етапи SDLC (Software development lifecycle) та є діаграмою Ганта.

Board для проекту CLP:

A screenshot of a cell phone

Description automatically generated

first Step

A screenshot of a cell phone

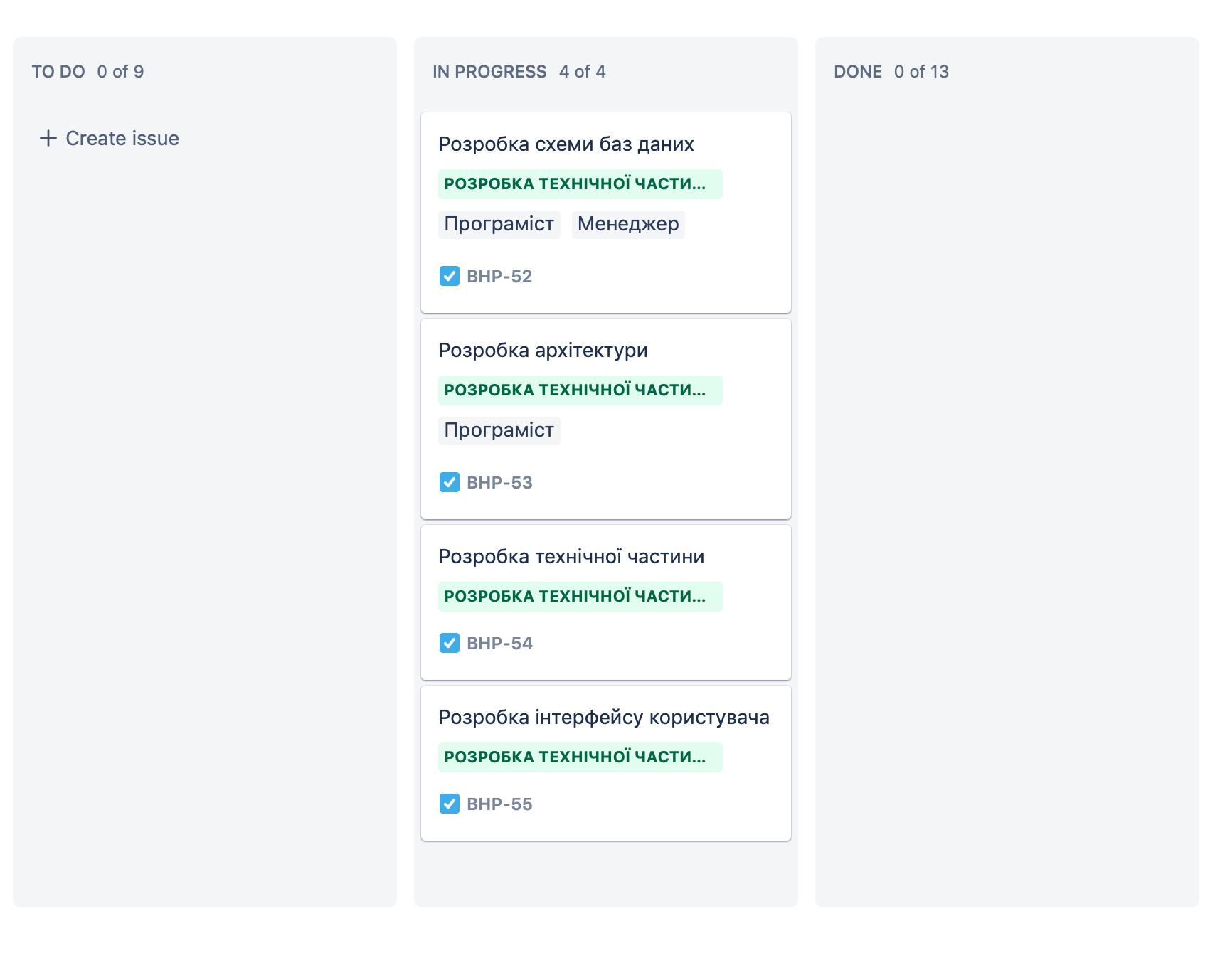
Description automatically generated

second Step

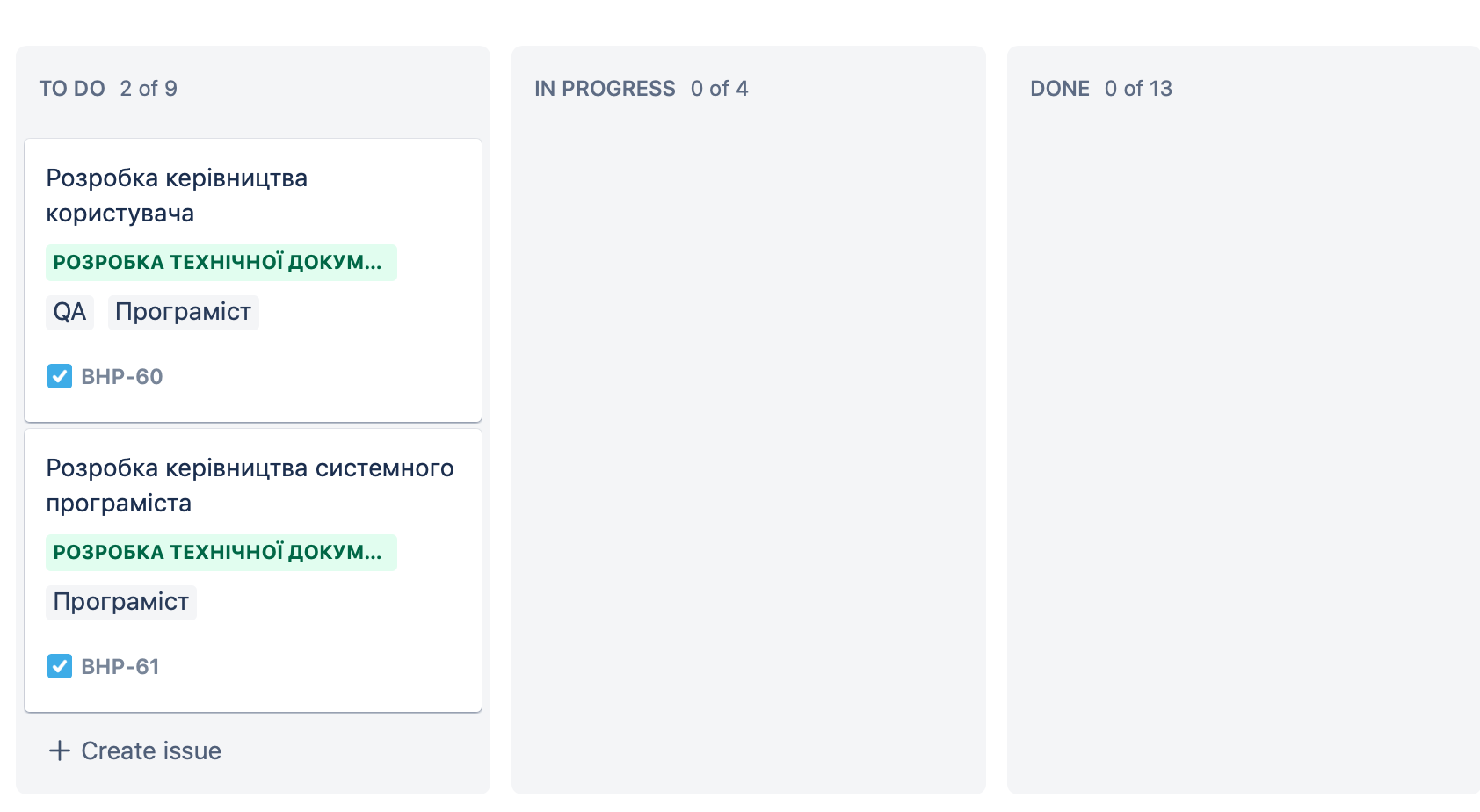
A screenshot of a cell phone

Description automatically generated

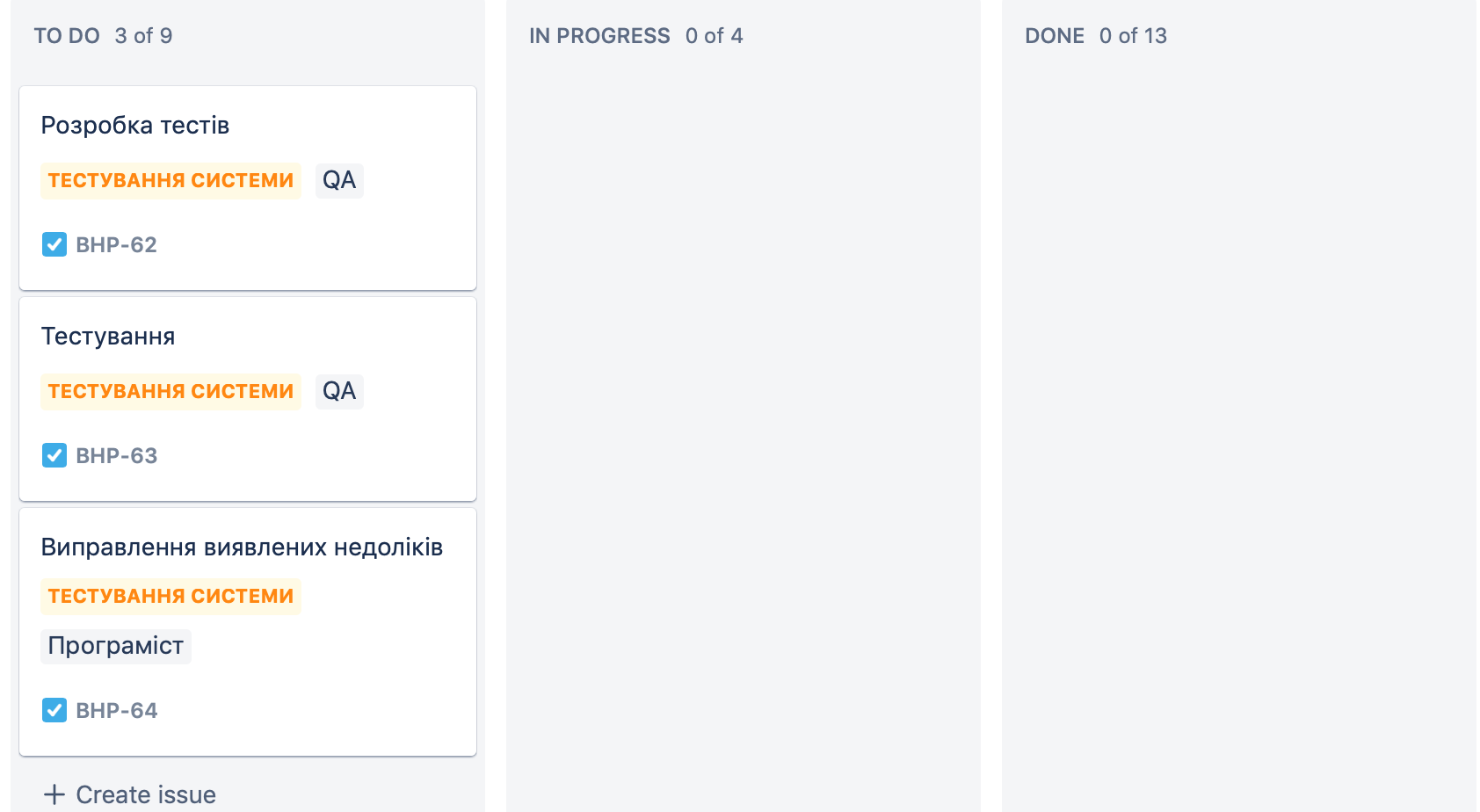
third Step



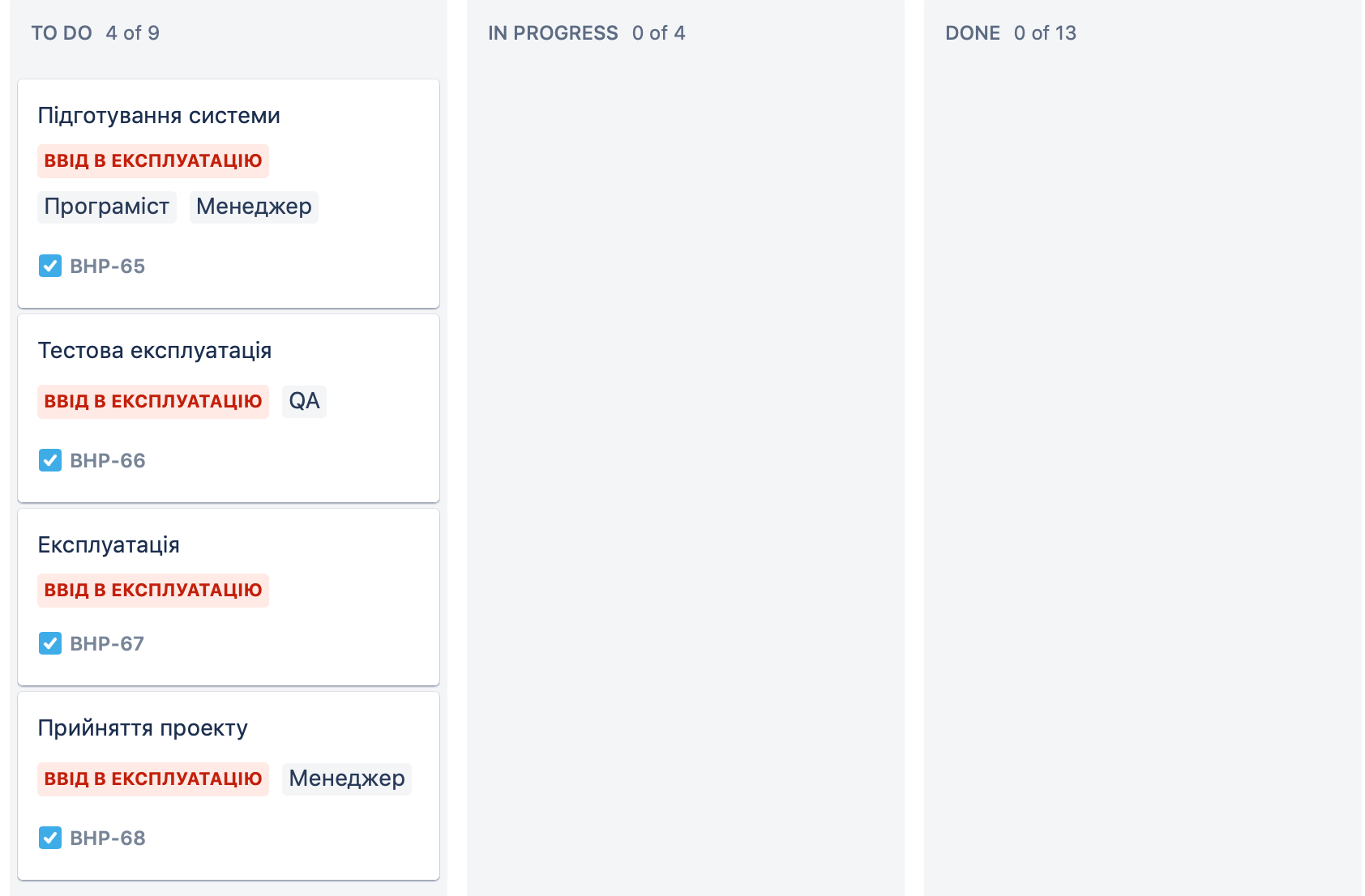
fourth Step



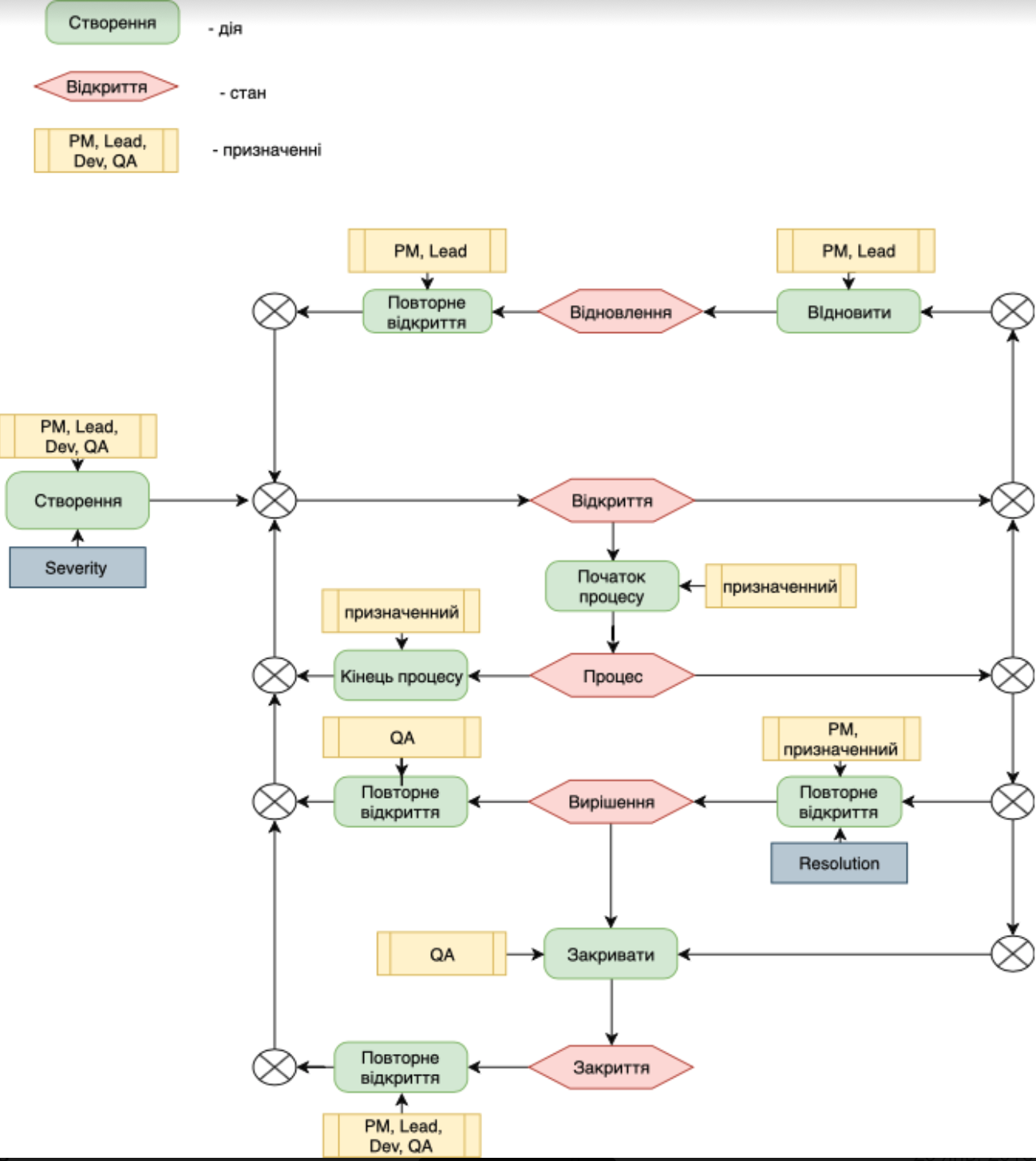
fifth Step



sixth Step



seventh Step

Діаграма життєвого циклу задачі (на прикладі дефекту):

При створенні завдання встановлюється Severity (тяжкість багу) для додатку та встановлення пріоритетності.

Resolution - опис причини повторного відкриття

Список готових частин інформаційної системи для кожного етапу:

|  |  |  |
| --- | --- | --- |
| **№** | **Навзва етапу / ітерація** | **Результат** |
| 1 | **Розробка концепції** | Остаточно вирішуємо, щодо концепції нашої системи шляхом аналізу актульності, вимог ринку та конкурентних пропозицій. Робимо один із 3 висновків: розробка нового додатку, покращення існуючого або відмовлення від ідеї. |
| 2 | **Розробка технічного завдання** | Необхідно визначити і задокументувати вимоги кінцевого користувача системи - в чому його очікування і як їх здійснити. Крім того, для проекту робиться техніко-економічне обґрунтування, яке з'ясовує, чи є проект організаційно, економічно, соціально, технологічно здійсненним. Дуже важливо підтримувати хороший рівень комунікації з замовниками, щоб переконатися, що у вас є чітке бачення кінцевого продукту і його функцій. |
| 3 | **Розробка ескізного проекту** | Після того, як досягнуто хороше розуміння вимог споживача. Визначаються елементи системи, компоненти, рівень безпеки, модулі, архітектуру, різні інтерфейси і типи даних, якими оперує система. Дизайн системи в загальних рисах може бути зроблений ручкою на листку паперу - він визначає, як система буде виглядати і як функціонувати. Потім робиться розширений, детальний дизайн, з урахуванням всіх функціональних і технічних вимог, як логічно, так і фізично. |
| 4 | **Розробка технічної частини** | На основі повного розумінням системних вимог і специфікацій відбувається реалізація додатку. Це і є власне процес розробки системи, коли дизайн системи вже повністю завершено і наочний. В життєвому циклі розробки системи саме тут пишеться код. На цій стадії система готова до реалізації технічної документації та подається на настпуний етап - тестування. |
| 5 | **Розробка технічної документації** | На основі готового ПП, розроблює технічну документацію до продукту |
| 6 | **Тестування системи** | Відбувається перевірка функціонування ПП на відповідність до ТЗ  Відбувається складання різних компонентів і підсистем в одну цілісну систему. Потім ми подаємо системі різні вхідні дані і аналізуємо вихід, поведінку і функціонування. Тестування стає все важливішим для задоволення споживача, при цьому воно не вимагає знань ні в КОДІНГ, ні в конфігурації обладнання, ні в дизайні. Тестування може виконуватися справжніми користувачами або спеціальною командою співробітників, також воно може бути систематичним і автоматизованим, з тим, щоб упевнитися, що актуальні результати роботи системи збігаються з передбаченими і бажаними. |
| 7 | **Ввід в експлуатацію** | Остаточно перевірений продукт в реальному середовищі та завершений проект |
| 8 | **Розробка концепції** | Підтримка додатку відповідно до ТЗ |

Список ризиків, їх вплив на проект та ймовірність виникнення.

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Ризики*** | ***Вплив на проект*** | ***Ймовірність виникнення*** |
| 1 | *Технологічні* |  |  |
| 1.1 | Поломка сервера | Відмова до створення нових записів та прочитання існуючих записів | Залежить від компанії сервера. Вважаємо мінімальною. |
| 1.2 | Збої на стороні провайдера | Можливо як і відмова функціональності повністю додатку, або не відповідає ТЗ (запис в неправильний каталог, інше). | Залежить від компанії сервера. Вважаємо мінімальною. |
| 2 | *Організаційні* |  |  |
| 2,1 | Неправильна організація робочого процесу | Нехватка часу призведе до пропущення дедлайнів та халатне виконання своїх обов’язків. | Вважаємо мінімальною. |
| 2.2 | Лінь працівників | Погано виконана робота та/або пропущення дедлайнів | Вважаємо мінімальною. |
| 2.3 | Нема кандидатів на найм | Немає людини - немає виконаного етапу | Вважаємо мінімальною. |
| 3 | *Ринкові* |  |  |
| 3.1 | Вартість сервера буде більшою на момент покупки за розраховану | Незмога повноціного виконання функцій додатку. | Вважаємо мінімальною. |
| 3,2 | Інфляція | Отриманні гроші знецінюються - дефіцит бюджету | Вважаємо мінімальною. |

|  |  |
| --- | --- |
| **Ризик** | **Захід** |
| Поломка сервера | Страховка Правильне техобслуговування Навчання програміста |
| Збої на стороні провайдера | Домовленість із запасним провайдером |
| Кваліфікація працівників | Відбір через співбесіду Заохочення плюшками |
| Нема кого наймати | Договір с HR-агентством |
| Вартість сервера більша за розраховану | Розраховувати із запасом |

**Висновок**

Програма для відстеження багів і завдань допомагає командам розробників виявляти, реєструвати і відслідковувати баги в створюваному ПО. Важливо забезпечити кожному члену команди можливість знаходити і фіксувати баги. Але ще важливіше призначати завдання по їх усуненню правильним учасникам команди в правильний час. Хороший інструмент для відстеження багів і завдань забезпечує для вашої команди єдине уявлення всіх елементів в беклозі незалежно від того, баг це або завдання, пов'язане з розробкою нових функцій. Єдиний надійний джерело інформації за всіма типами задач допомагає командам розставляти пріоритети, орієнтуючись на основні стратегічні цілі, і при цьому постійно працювати на забезпечення потреб клієнтів.