**ПланТестуванняЖ** TaskFlow

Підготовлено**:**

Hnativ Sviatoslav

03.04.2024

1. **INTRODUCTION**

Додаток TaskFlow - це комплексне та інтуїтивно зрозуміле рішення для управління завданнями, ретельно розроблене для оптимізації робочих процесів та підвищення продуктивності. По суті, TaskFlow слугує централізованим центром, що дозволяє користувачам легко створювати, розподіляти, відстежувати та визначати пріоритети завдань у різних командах та проектах. Основні функції та можливості TaskFlow включають:

1. Легке створення та призначення завдань: За допомогою TaskFlow користувачі можуть легко створювати завдання, призначати їх конкретним членам команди та встановлювати чіткі терміни виконання. Інтуїтивно зрозумілий інтерфейс гарантує, що створення завдань є простим процесом, що сприяє ефективному делегуванню завдань і відповідальності за них.
2. Динамічне відстеження завдань: TaskFlow полегшує моніторинг прогресу завдань у режимі реального часу, пропонуючи користувачам неперевершену видимість оновлень статусу, коментарів і пов'язаних з ними вкладень. Завдяки настроюваним інформаційним панелям і системам сповіщень користувачі можуть залишатися в курсі розвитку завдань і вживати проактивних заходів для усунення будь-яких вузьких місць або затримок.
3. Складна пріоритизація завдань: TaskFlow дозволяє користувачам визначати пріоритетність завдань на основі різних критеріїв, таких як терміновість, важливість або терміни виконання проекту. Платформа надає надійні функції визначення пріоритетів, що дозволяє користувачам ефективно розподіляти ресурси і гарантувати, що критичні завдання отримають увагу, на яку вони заслуговують.
4. Безперешкодна співпраця: Співпраця лежить в основі TaskFlow, сприяючи синергії та згуртованості між членами команди. Додаток полегшує безперешкодне спілкування, обмін файлами та спільну роботу над завданнями, сприяючи обміну знаннями та колективному вирішенню проблем.
5. Глибокі звіти та аналітика: TaskFlow пропонує комплексні можливості звітності та аналітики, що дозволяють користувачам отримувати цінну інформацію про виконання завдань, використання ресурсів і терміни виконання проектів. За допомогою настроюваних звітів та інтерактивних інформаційних панелей користувачі можуть виявляти тенденції, відстежувати ключові показники та приймати рішення на основі даних для досягнення організаційного успіху.

Таким чином, TaskFlow є універсальним і незамінним інструментом для сучасного бізнесу, який прагне впорядкувати процеси управління завданнями, покращити співпрацю в команді та підвищити операційну ефективність. Використовуючи можливості TaskFlow, організації можуть оптимізувати розподіл ресурсів, прискорити реалізацію проектів і впевнено досягати своїх стратегічних цілей.

1. **OBJECTIVES AND TASKS**
   1. **Objectives**

Генеральний план тестування (ГПТ) слугує комплексним документом, який окреслює основні цілі та стратегії тестування додатку TaskFlow. Його основна мета - надати дорожню карту для всього процесу тестування, забезпечуючи ретельність, ефективність і узгодженість тестової діяльності. Цілі, що підтримуються генеральним планом тестування, включають

1. Визначення завдань та обов'язків: ССП окреслює конкретні завдання та обов'язки кожної особи або команди, залученої до процесу тестування. Сюди входять такі ролі, як тест-менеджери, тест-аналітики, розробники та зацікавлені сторони, разом з їхніми відповідними обов'язками та зобов'язаннями. Наприклад, тест-менеджери можуть відповідати за нагляд за всім процесом тестування, координацію ресурсів та звітування про прогрес перед зацікавленими сторонами проекту. Тестовим аналітикам може бути доручено створення планів тестування, розробка тестових кейсів, виконання тестів і документування результатів. Від розробників може знадобитися усунення дефектів, виявлених під час тестування, і забезпечення своєчасного вирішення проблем. Чітко визначаючи завдання та обов'язки, ССП гарантує, що всі зацікавлені сторони розуміють свої ролі та роблять ефективний внесок у процес тестування.
2. Засіб комунікації: ССП слугує центральним засобом комунікації з усіх питань, пов'язаних з тестуванням. Він встановлює канали для поширення інформації, обміну оновленнями та координації діяльності між членами команди та зацікавленими сторонами. Це може включати регулярні звіти про стан, зустрічі та перегляд документації. Наприклад, ССП може передбачати щотижневі наради для обговорення прогресу, вирішення проблем та узгодження пріоритетів. Він також може містити протоколи для спілкування між командою тестувальників та іншими зацікавленими сторонами проекту, такими як розробники, менеджери продуктів і кінцеві користувачі. Сприяючи прозорому спілкуванню, ССП гарантує, що всі дотримуються цілей і термінів тестування.
3. Угода про рівень обслуговування (SLA): ССП слугує фактичною угодою про рівень обслуговування (SLA) між командою тестування та іншими зацікавленими сторонами проекту. Він встановлює стандарти якості, методології тестування та показники ефективності, які будуть керувати процесом тестування. Наприклад, ССП може визначати критерії для тестового покриття, рівні серйозності дефектів та критерії прийняття. Він також може окреслити цілі рівня обслуговування для ключових видів діяльності з тестування, таких як виконання тестів, усунення дефектів та звітування. Встановлюючи чіткі очікування і результати, ССП допомагає зменшити непорозуміння і гарантує, що зусилля з тестування відповідають загальним цілям і вимогам проекту.
4. Управління ризиками: ССП включає стратегії управління ризиками для виявлення, оцінки та пом'якшення потенційних ризиків, які можуть вплинути на процес тестування або якість заявки. Він містить плани дій у надзвичайних ситуаціях, заходи з пом'якшення наслідків і процедури ескалації для вирішення будь-яких непередбачуваних проблем або перешкод, що виникають під час тестування. Наприклад, ССП може визначати такі ризики, як обмеженість ресурсів, технічні залежності або зміни у вимогах. Він може визначити пріоритетність ризиків на основі серйозності, ймовірності та впливу на цілі тестування. Завдяки проактивному управлінню ризиками, ССП допомагає мінімізувати збої та забезпечує своєчасне завершення діяльності з тестування.
5. 5. Розподіл ресурсів: ССП містить настанови щодо розподілу ресурсів, зокрема людських ресурсів, інструментів тестування та вимог до інфраструктури. Він окреслює оптимальне використання ресурсів для максимізації ефективності та результативності тестування з дотриманням бюджетних обмежень і термінів проекту. Наприклад, ССП може визначати розподіл тестових середовищ, ліцензій на програмне забезпечення та обладнання для тестування. Він також може визначати штатний розклад, вимоги до навичок і потреби в навчанні для команди тестувальників. Узгоджуючи розподіл ресурсів з пріоритетами та цілями тестування, ССП гарантує, що зусилля з тестування не відстають від графіка і не виходять за рамки бюджету.
6. Стандарти документації: ССП встановлює стандарти і шаблони документації, які будуть використовуватися протягом усього процесу тестування. Сюди входять плани тестування, тестові кейси, тестові сценарії, звіти про дефекти та інша відповідна документація. Наприклад, ССП може визначати структуру і формат тестових планів, включаючи розділи для цілей, обсягу, підходу і результатів. Він також може визначати зміст і структуру тестових кейсів, включаючи етапи тестування, очікувані результати і критерії прийнятності. Впроваджуючи узгоджену практику документування, ССП забезпечує чіткість, точність і повноту документування діяльності з тестування, його результатів і висновків.

Генеральний план тестування (ГПТ) відіграє ключову роль у забезпеченні успіху процесу тестування додатку TaskFlow. Визначаючи чіткі цілі та стратегії, MTP слугує керівною основою для всього процесу тестування, сприяючи ретельності, ефективності та послідовності. Завдяки визначеним цілям, ССП встановлює дорожню карту для команди тестування, окреслюючи конкретні завдання та обов'язки для кожної особи або команди, що беруть участь у тестуванні. Чіткий розподіл ролей та обов'язків забезпечує ефективну координацію та співпрацю між членами команди, що призводить до підвищення продуктивності та підзвітності. Крім того, ССП слугує важливим засобом комунікації, сприяючи створенню прозорих каналів зв'язку між зацікавленими сторонами. Регулярне оновлення, зустрічі та перегляд документації за сприяння ССП гарантують, що всі будуть поінформовані та узгоджені з цілями і термінами тестування. Як фактична угода про рівень обслуговування, ССП встановлює стандарти якості, методології тестування та показники ефективності, які регулюють процес тестування. Це гарантує, що зусилля з тестування узгоджуються із загальними цілями та вимогами проекту, зменшуючи непорозуміння та гарантуючи, що результати відповідають очікуванням. Крім того, ССП включає стратегії управління ризиками для виявлення, оцінки та пом'якшення потенційних ризиків, які можуть вплинути на процес тестування або якість заявки. Завдяки проактивному управлінню ризиками, ССП допомагає мінімізувати збої та забезпечує своєчасне завершення тестування. Керівні принципи розподілу ресурсів, передбачені ССП, забезпечують оптимальне використання ресурсів з дотриманням бюджетних обмежень і термінів проекту. Це гарантує, що зусилля з тестування не виходять за рамки запланованого плану і не виходять за рамки бюджету, забезпечуючи максимальну ефективність і результативність. Нарешті, ССП встановлює стандарти і шаблони документації, які повинні використовуватися протягом усього процесу тестування, забезпечуючи чіткість, точність і повноту документування діяльності з тестування, його результатів і висновків. Насамкінець, Генеральний план тестування слугує всеосяжним керівництвом та основою управління процесом тестування, охоплюючи всі аспекти планування, виконання та оцінки. Його ретельний підхід гарантує, що додаток TaskFlow пройде ретельне тестування, в результаті чого буде створено високоякісний, надійний і зручний для користувача продукт.

* 1. **Tasks**

1. Тестові завдання:
   1. Планування тестування: Розробка детального плану тестування із зазначенням цілей, обсягу, підходу та необхідних ресурсів. Сюди входить визначення методологій, інструментів і методів тестування, які будуть використовуватися протягом усього процесу тестування. План тестування також повинен встановлювати критерії покриття тестом і критерії прийняття.
   2. Розробка тестових кейсів: Створіть комплексні тестові кейси, що охоплюють всі можливості та функції додатку TaskFlow, включаючи позитивні та негативні сценарії. Тестові кейси повинні бути розроблені для перевірки кожної вимоги та історії користувача, забезпечуючи повне охоплення всіх можливих варіантів використання.
   3. Налаштування тестового середовища: Налаштуйте необхідне тестове середовище, включаючи обладнання, програмне забезпечення та мережеві конфігурації, щоб відтворити виробничі умови. Сюди входить встановлення та налаштування програми TaskFlow, а також будь-яких допоміжних інструментів або систем, необхідних для тестування.
   4. Виконання тестів: Виконання тестових кейсів відповідно до попередньо визначеного плану тестування, забезпечуючи ретельне покриття всіх тестових сценаріїв. Виконання тестів передбачає дотримання кроків, описаних у тестових кейсах, запис результатів тестування та фіксацію будь-яких відхилень або дефектів, виявлених під час тестування.
   5. Ідентифікація дефектів: Ідентифікуйте та задокументуйте будь-які дефекти або невідповідності, виявлені під час виконання тесту, включаючи кроки для відтворення та оцінки серйозності. Дефекти слід реєструвати в системі відстеження дефектів, надаючи детальний опис і додаючи відповідну документацію або скріншоти.
   6. Перевірка усунення дефектів: Перевірте усунення дефектів, про які повідомлялося, повторно протестувавши відповідні функціональні можливості, щоб переконатися, що вони усунуті належним чином. Це передбачає виконання тестових кейсів, пов'язаних з вирішеними дефектами, і перевірку того, що проблеми були вирішені задовільно.
   7. Регресійне тестування: Проведення регресійного тестування, щоб переконатися, що нові зміни або виправлення не призведуть до небажаних побічних ефектів або регресу в існуючих функціях. Регресійне тестування передбачає повторне виконання раніше виконаних тестових кейсів, щоб переконатися, що існуючі функціональні можливості залишаються недоторканими після внесення змін.
   8. Тестування продуктивності: Виконуйте тестування продуктивності, щоб оцінити швидкість відгуку, масштабованість і стабільність роботи програми TaskFlow за різних умов навантаження. Тестування продуктивності включає в себе моделювання взаємодії користувачів і моніторинг показників продуктивності системи, таких як час відгуку, пропускна здатність і використання ресурсів.
   9. Навантажувальне тестування: Проведення стрес-тестування для оцінки стійкості та надійності програми, піддаючи її екстремальним робочим навантаженням. Навантажувальне тестування передбачає виведення системи за межі її нормальної роботи, щоб виявити потенційні точки відмови або погіршення продуктивності.
   10. Підтримка тестування користувацького сприйняття (UAT): Допомагайте зацікавленим сторонам у проведенні користувацького тестування, щоб підтвердити, що TaskFlow відповідає вимогам та очікуванням користувачів. Це включає в себе надання рекомендацій щодо виконання тестів, збір відгуків від користувачів і полегшення комунікації між зацікавленими сторонами і командами розробників.
2. Після тестові завдання:
   1. .Звіт про результати тестування: Складіть комплексний звіт про результати тестування, в якому детально опишіть проведені тестові заходи, включаючи результати тестування, виявлені дефекти та загальну оцінку якості додатку. Звіт про результати тестування повинен давати уявлення про ефективність тестування та будь-які проблемні області, які можуть потребувати подальшої уваги.
   2. Оновлення документації: Оновлення тестової документації, включаючи плани тестування, тестові кейси та інші відповідні артефакти, щоб відобразити будь-які зміни або уроки, отримані в процесі тестування. Це включає в себе перегляд документації для врахування відгуків, усунення недоліків і забезпечення відповідності вимогам проєкту.
3. Завдання повідомлення про проблеми:
   1. Звітування про дефекти: Документування дефектів, виявлених під час тестування в системі відстеження дефектів, включаючи докладні описи, кроки для відтворення та оцінки серйозності. Дефекти повинні бути пріоритезовані на основі серйозності та впливу на функціональність програми, що полегшить ефективне вирішення проблем командами розробників.
   2. Сортування дефектів: Визначення пріоритетів дефектів на основі серйозності, впливу на функціональність програми та бізнеспріоритетів, щоб полегшити їх ефективне усунення. Сортування дефектів включає в себе оцінку серйозності та впливу повідомлених дефектів і визначення їх пріоритетності для усунення на основі заздалегідь визначених критеріїв. 3.3.Контроль за усуненням дефектів: Спілкуйтеся з командами розробників, щоб забезпечити своєчасне усунення виявлених дефектів, надаючи додаткову інформацію або роз'яснення, якщо це необхідно. Контроль усунення дефектів включає відстеження статусу повідомлених дефектів, ескалацію невирішених проблем за необхідності та перевірку задовільного впровадження виправлень.
4. Комунікаційні завдання:
   1. Наради щодо статусу.
   2. Перевірка документації: Співпрацюйте зі стейкхолдерами для перегляду та валідації тестової документації, забезпечуючи точність, повноту та відповідність вимогам проєкту. Перевірка документації дає можливість отримати зворотній зв'язок, вирішити проблеми і переконатися, що артефакти тестування відповідають стандартам якості.
   3. Комунікація зі стейкхолдерами.
5. Завдання документації та звітності:
   1. Перегляд плану тестування: Перегляд і доопрацювання плану тестування з урахуванням відгуків зацікавлених сторін і забезпеченням відповідності цілям і вимогам проекту. Перевірка плану тестування включає перевірку точності та повноти плану тестування, вирішення будь-яких проблем або розбіжностей, а також отримання схвалення від відповідних зацікавлених сторін.
   2. Перегляд тестових кейсів: Перегляньте та доопрацюйте тестові кейси, переконавшись, що вони точно відображають тестові сценарії, очікувані результати та критерії прийнятності. Перевірка тестових кейсів включає перевірку ефективності та покриття тестових кейсів, усунення будь-яких двозначностей або невідповідностей, а також отримання схвалення від відповідних зацікавлених сторін.
   3. Звіт про виконання тесту: Створюйте звіти про виконання тестів, які підсумовують результати виконання тестів, включаючи статус проходження/непроходження, метрики дефектів і аналіз покриття. Звіти про виконання тестів надають зацікавленим сторонам інформацію про якість і надійність програми TaskFlow, допомагаючи приймати рішення і визначати пріоритети подальших дій.

Загалом, детальний розподіл завдань, представлений у Плані тестування, слугує дорожньою картою для команди тестування, спрямовуючи їхні зусилля та гарантуючи, що діяльність з тестування проводиться методично та всебічно. Дотримуючись цих завдань, команда тестувальників може допомогти забезпечити успіх програми TaskFlow і сприяти досягненню цілей проекту

1. **SCOPE**

**General**

Цей розділ містить детальний огляд того, що тестується в рамках Плану тестування. Він охоплює всі функції програми TaskFlow, включаючи існуючі інтерфейси та інтеграцію всіх функцій. Тестування спрямоване на забезпечення функціональності, надійності та продуктивності програми TaskFlow в різних сценаріях і варіантах використання.

1. Функціональне тестування:
   1. Функціональність програми TaskFlow буде ретельно протестована, щоб переконатися, що всі функції та можливості працюють належним чином. Це включає тестування створення, призначення, відстеження та визначення пріоритетів завдань, а також будь-яких додаткових функцій, таких як спільний доступ до файлів, коментування та спільна робота над завданнями.
   2. Кожна окрема функція в TaskFlow буде перевірена, щоб переконатися, що вона поводиться так, як очікується за нормальних умов роботи. Це передбачає перевірку входів, виходів і очікуваної поведінки на відповідність попередньо визначеним вимогам і очікуванням користувача.
2. Тестування інтерфейсу:
   1. Тестування інтерфейсу фокусується на перевірці взаємодії між різними компонентами і модулями в додатку TaskFlow. Сюди входить тестування користувацького інтерфейсу (UI) на зручність, узгодженість і швидкість реагування на різних пристроях і розмірах екранів.
   2. Крім того, буде проведено інтеграційне тестування для забезпечення безперебійного зв'язку та обміну даними між TaskFlow і будь-якими зовнішніми системами або сторонніми інтеграціями. Це передбачає перевірку узгодженості даних, обробку помилок і заходів безпеки, впроваджених під час передачі даних.
3. Інтеграція функцій:
   1. Інтеграційне тестування оцінює інтеграцію всіх функцій в додатку TaskFlow, щоб переконатися, що вони гармонійно працюють разом, виконуючи вимоги користувачів і бізнес-цілі.
   2. Це передбачає тестування наскрізних сценаріїв, які охоплюють кілька функцій і модулів, імітуючи реальні взаємодії користувачів і робочі процеси. Інтеграційне тестування має на меті виявити будь-які потенційні проблеми або невідповідності, які можуть виникнути під час взаємодії різних функцій між собою.
4. Граничне тестування:
   1. Граничне тестування буде виконано для перевірки поведінки TaskFlow на межі його операційних обмежень. Це включає тестування здатності програми обробляти максимальні і мінімальні вхідні значення, а також граничні умови, такі як верхня і нижня межі для пріоритетів завдань, дедлайнів та інших параметрів.
   2. Граничне тестування допомагає переконатися, що TaskFlow може впоратися з екстремальними сценаріями, не зіткнувшись з несподіваними помилками або пошкодженням даних.
5. Обробка помилок і тестування відновлення:
   1. Тестування обробки помилок і відновлення дозволить оцінити здатність TaskFlow виявляти, повідомляти і відновлюватися після помилок. Це включає тестування повідомлень про помилки, механізмів обробки помилок і процедур відновлення, щоб гарантувати, що користувачі отримають інформативний зворотний зв'язок і зможуть відновити нормальну роботу без втрати або пошкодження даних.
   2. Крім того, для імітації ненормальних умов і перевірки того, що TaskFlow реагує належним чином за несприятливих обставин, можуть бути використані методи введення помилок.
6. Тестування на сумісність:
   1. Тестування сумісності буде проведено для перевірки сумісності TaskFlow з різними операційними системами, веб-браузерами та пристроями, які зазвичай використовуються цільовою аудиторією. Це включає тестування сумісності з різними версіями Windows, macOS, iOS, Android та популярними веб-браузерами, такими як Chrome, Firefox, Safari та Edge.
   2. Тестування сумісності має на меті забезпечити узгоджену роботу користувачів у різних середовищах і на різних платформах, мінімізуючи проблеми сумісності та максимізуючи доступність для всіх користувачів.

Таким чином, загальний розділ плану тестування описує комплексний підхід до тестування додатку TaskFlow, що охоплює всі функції, інтерфейси та аспекти інтеграції. Ретельно тестуючи кожен компонент і сценарій, команда тестувальників прагне підтвердити функціональність, надійність і зручність використання TaskFlow, щоб надати кінцевим користувачам високоякісний продукт.

**Tactics**

Відповідно до цілей, викладених в розділі "Обсяг", в розділі "Тактика" Плану тестування детально описані конкретні стратегії і методології, які будуть використані для досягнення цілей тестування. Ця тактика слугує дорожньою картою для виконання тестової діяльності, описаної в розділі "Обсяг", забезпечуючи систематичний та комплексний підхід до перевірки додатку TaskFlow. Тепер давайте заглибимося в детальну тактику, яка буде використана для досягнення кожної з цілей:

1. Функціональне тестування:
   1. Розробка тестових кейсів: Розробіть комплексні тестові кейси, що охоплюють всі функції додатку TaskFlow. Тестові кейси будуть розроблені для перевірки кожної окремої функції, включаючи входи, виходи та очікувану поведінку.
   2. Виконання тестів: Виконання тестових кейсів відповідно до попередньо визначеного плану тестування, забезпечуючи ретельне покриття всіх функцій. Тестувальники систематично проходять кожен тестовий кейс, записуючи результати та виявляючи будь-які відхилення від очікуваної поведінки.
   3. Ідентифікація дефектів: Задокументуйте будь-які дефекти або невідповідності, виявлені під час виконання тесту. Дефекти будуть зареєстровані в системі відстеження дефектів з детальним описом і додаванням відповідної документації або скріншотів.
   4. Перевірка усунення дефектів: Перевірте усунення дефектів, про які повідомлялося, повторно протестувавши відповідні функції, щоб переконатися, що вони усунуті належним чином. Це передбачає виконання тестових кейсів, пов'язаних з вирішеними дефектами, і перевірку того, що проблеми були вирішені задовільно.
2. Тестування інтерфейсу:
   1. Тестування інтерфейсу користувача (UI): Проведіть тестування інтерфейсу користувача, щоб забезпечити зручність, узгодженість і швидкість реагування на різних пристроях і розмірах екранів. Тестувальники оцінять макет, елементи дизайну та навігацію додатку TaskFlow.
   2. Тестування інтеграції: Перевірте безперебійний зв'язок і обмін даними між TaskFlow і будь-якими зовнішніми системами або сторонніми інтеграціями. Тестувальники перевірять узгодженість даних, обробку помилок і заходи безпеки, застосовані під час передачі даних.
   3. Тестування сумісності: Перевірте сумісність TaskFlow з різними операційними системами, веб-браузерами та пристроями, які зазвичай використовує ваша цільова аудиторія. Тестувальники оцінять поведінку та продуктивність додатку в різних середовищах і на різних платформах.
3. Інтеграція функцій:
   1. Наскрізне тестування: Проведіть наскрізне тестування, щоб перевірити інтеграцію всіх функцій в TaskFlow. Тестувальники імітуватимуть реальні взаємодії користувачів і робочі процеси, щоб переконатися, що різні функції працюють злагоджено, виконуючи вимоги користувачів і бізнес-цілі.
   2. Граничне тестування: Проведіть граничне тестування, щоб перевірити поведінку TaskFlow на межі його операційних меж. Тестувальники оцінять здатність програми обробляти максимальні і мінімальні вхідні значення, а також граничні умови, такі як верхня і нижня межі пріоритетів завдань, дедлайни та інші параметри.
4. Обробка помилок і тестування відновлення:
   1. Тестування ін'єкції помилок: Змоделюйте аномальні умови і перевірте, чи правильно TaskFlow реагує на помилки. Тестувальники навмисно вводять помилки в додаток, щоб оцінити його механізми обробки помилок і процедури відновлення.
   2. Оцінка відгуків користувачів: Оцініть чіткість та ефективність повідомлень про помилки та підказок користувача, які надає TaskFlow. Тестувальники повинні переконатися, що користувачі отримують інформативний зворотній зв'язок і рекомендації щодо усунення помилок, які виникають під час роботи з додатком.
5. Тестування на сумісність:
   1. Тестування платформи: Перевірте сумісність TaskFlow з різними операційними системами, включаючи Windows, macOS, iOS та Android. Тестувальники оцінять поведінку та продуктивність програми на кожній з підтримуваних платформ.
   2. Тестування сумісності з браузерами: Перевірте сумісність TaskFlow з популярними веб-браузерами, такими як Chrome, Firefox, Safari та Edge. Тестувальники переконаються, що додаток функціонує коректно і послідовно відображається в різних середовищах браузерів.

Ця тактика описує конкретні процедури і методології, які будуть використані для досягнення цілей тестування, зазначених у розділі "Обсяг" Плану тестування. Систематично дотримуючись цієї тактики, команда тестувальників прагне ретельно оцінити функціональність, надійність і продуктивність додатку TaskFlow, забезпечуючи високу якість продукту, який відповідає вимогам і очікуванням користувачів.

1. **TESTING STRATEGY**

Стратегія тестування програми TaskFlow передбачає комплексний підхід для забезпечення функціональності, надійності, продуктивності та зручності використання програмного забезпечення. Кожна основна група функцій або комбінацій функцій буде проходити специфічні методики тестування, адаптовані до їхніх унікальних характеристик і вимог.

* 1. **Alpha Testing (Unit Testing)**

**Definition:**

1. Бажана повнота: Юніт-тестування має на меті досягти мінімального рівня повноти, гарантуючи, що кожна одиниця коду, така як функції або методи, буде ретельно протестована для перевірки її коректності та функціональності.
2. Методи оцінки повноти: Повнота модульного тестування буде оцінюватися на основі метрик покриття коду, таких як покриття операторів, покриття гілок і покриття шляхів. Ці метрики допоможуть визначити, які оператори та шляхи коду були виконані під час тестування.
3. Додаткові критерії завершення: На додаток до метрик покриття коду, критерії завершення модульного тестування можуть включати частоту помилок, щільність дефектів, а також дотримання стандартів кодування та найкращих практик.
4. Методи відстеження вимог: Матриці відстеження вимог (RTM) будуть використовуватися для відстеження кожного тестового кейсу до конкретних вимог або користувацьких історій, які він перевіряє.

**Participants:**

Відповідальність за модульне тестування несе команда розробників, зокрема, окремі розробники, відповідальні за написання коду, що тестується.

**Methodology:**

1. Авторство тестових скриптів: Розробники будуть писати тестові скрипти для модульного тестування як частину процесу кодування. Кожен розробник буде відповідати за створення модульних тестів для написаного ним коду, забезпечуючи повне покриття всіх функціональних можливостей.
2. Послідовність подій: Послідовність подій для модульного тестування зазвичай слідує за процесом розробки. Коли розробники пишуть код, вони паралельно розробляють модульні тести для перевірки функціональності свого коду. Після завершення роботи над кодом розробники виконують відповідні модульні тести, щоб переконатися в його коректності.
3. Діяльність з тестування: Юніт-тестування відбуватиметься в середовищі розробки з використанням фреймворків для юніт-тестування, таких як JUnit, NUnit або Pytest. Розробники будуть виконувати модульні тести локально на своїх робочих машинах або за допомогою автоматизованих конвеєрів збірки. Будь-які збої або дефекти, виявлені під час модульного тестування, будуть оперативно усунені відповідними розробниками до того, як код буде інтегрований в основну кодову базу.

Дотримуючись такого підходу до модульного тестування, команда розробників прагне забезпечити надійність і коректність окремих одиниць коду в додатку TaskFlow, закладаючи міцний фундамент для наступних етапів тестування.

* 1. **System and Integration Testing**

**Definition:**

1. Бажана комплексність: Системне та інтеграційне тестування має на меті всебічну перевірку взаємодії та сумісності між різними компонентами та модулями додатку TaskFlow.
2. Методи оцінки комплексності: Комплексність системного та інтеграційного тестування буде оцінюватися на основі метрик тестового покриття, включаючи функціональне покриття, інтеграційне покриття та покриття сценаріїв. Ці показники вказуватимуть на ступінь, до якого були протестовані різні функціональні можливості та точки інтеграції.
3. Додаткові критерії завершення: Критерії завершення системного та інтеграційного тестування можуть включати відсутність критичних дефектів, що впливають на функціональність системи, дотримання вимог до продуктивності та масштабованості, а також успішне виконання наскрізних тестових сценаріїв.
4. Методи відстеження вимог: Матриці відстеження вимог (RTM) будуть використовуватися для відстеження кожного тестового сценарію і тестового кейсу до конкретних функціональних вимог, історій користувачів або варіантів використання, які вони перевіряють

**Participants:**

За системне та інтеграційне тестування відповідає команда тестувальників на чолі з тест-менеджером або тест-лідером. Крім того, розробники можуть бути залучені до підтримки інтеграційного тестування, зокрема, для вирішення проблем та дефектів інтеграції

**Methodology:**

1. Розробка тестових сценаріїв: Тестові сценарії будуть розроблені для моделювання наскрізних взаємодій користувачів і робочих процесів, що охоплюють різні функції та модулі програми TaskFlow. Ці сценарії будуть базуватися на функціональних вимогах, історіях користувачів та варіантах використання, забезпечуючи всебічне охоплення функціональності системи.
2. Підготовка тестових даних: Тестові дані будуть підготовлені для підтримки виконання тестових сценаріїв, включаючи як правильні, так і неправильні вхідні дані. Тестові дані будуть розроблені таким чином, щоб охоплювати різні сценарії використання та граничні випадки, забезпечуючи надійне тестування поведінки системи за різних умов.
3. Налаштування тестового середовища: Буде створено спеціальне тестове середовище, яке максимально точно відтворюватиме виробниче середовище. Це середовище включатиме необхідне обладнання, програмне забезпечення та мережеві конфігурації для підтримки системного та інтеграційного тестування.
4. Виконання тестів: Тестові сценарії будуть виконуватися в тестовому середовищі відповідно до заздалегідь визначених тестових процедур і скриптів. Тестувальники перевірятимуть поведінку програми TaskFlow, включаючи її інтерфейси та взаємодію із зовнішніми системами або сторонніми інтеграціями.
5. Управління дефектами: Будь-які дефекти або проблеми, виявлені під час системного та інтеграційного тестування, будуть реєструватися в системі відстеження дефектів з детальним описом, кроками для відтворення та оцінкою серйозності. Дефектам буде визначено пріоритети та призначено команди розробників для усунення, з відповідними подальшими діями та перевіркою виправлень.
6. Регресійне тестування: Регресійне тестування буде проводитися для того, щоб переконатися, що існуючі функціональні можливості залишаються недоторканими після системних змін або оновлень. Набори регресійних тестів будуть виконуватися регулярно, щоб виявити будь-які регресії або ненавмисні побічні ефекти, що виникли в результаті системних модифікацій.
7. Тестування продуктивності: Тестування продуктивності буде проведено для оцінки швидкості реагування, масштабованості та стабільності додатку TaskFlow за різних умов навантаження. Показники продуктивності будуть зібрані та проаналізовані, щоб визначити вузькі місця та області для оптимізації.

Дотримуючись цієї методології системного та інтеграційного тестування, команда тестувальників має на меті перевірити поведінку та продуктивність додатку TaskFlow у всій його системній архітектурі, забезпечуючи безперебійну взаємодію між різними компонентами та модулями.

* 1. **Performance and Stress Testing**

**Definition:**

1. Бажана комплексність: Тестування продуктивності та стрес-тестування має на меті всебічно оцінити реакцію, масштабованість та стабільність додатку TaskFlow за нормальних та екстремальних умов навантаження.
2. Методи оцінки комплексності: Комплексність при тестуванні продуктивності та стрес-тестуванні буде оцінюватися на основі показників продуктивності, таких як час відгуку, пропускна спроможність та використання ресурсів. Ці показники вказуватимуть на здатність додатку відповідати вимогам до продуктивності та обробляти очікувані обсяги робочого навантаження.
3. Додаткові критерії завершення: Критерії завершення тестування продуктивності та стрес-тестування можуть включати відповідність попередньо визначеним критеріям продуктивності, підтримання прийнятного часу відгуку в умовах пікового навантаження та демонстрацію плавної деградації під навантаженням. Методи відстеження вимог: Вимоги до продуктивності та бенчмарки будуть відстежуватися за конкретними сценаріями тестування продуктивності та метриками, щоб гарантувати, що тестування адекватно охоплює всі аспекти, пов'язані з продуктивністю.

**Participants:**

Відповідальність за тестування продуктивності та стрес-тестування покладається на команду тестувальників за підтримки інженерів з продуктивності або фахівців, які мають досвід роботи з методологіями та інструментами тестування продуктивності.

**Methodology:**

1. Розробка сценаріїв тестування продуктивності: Сценарії тестування продуктивності будуть розроблені для імітації різних взаємодій користувачів і моделей робочого навантаження, очікуваних у виробництві. Ці сценарії будуть базуватися на профілях використання, бізнес-процесах та очікуваних обсягах трафіку користувачів.
2. Підготовка тестових даних: Набори тестових даних будуть підготовлені для підтримки виконання сценаріїв тестування продуктивності, включаючи як реалістичні, так і синтетичні дані. Тестові дані будуть розроблені таким чином, щоб представляти типові сценарії використання та граничні випадки, забезпечуючи точну симуляцію виробничого навантаження.
3. Налаштування тестового середовища: Буде створено спеціальне середовище для тестування продуктивності, яке максимально точно відтворюватиме виробниче середовище. Це середовище включатиме необхідне обладнання, програмне забезпечення та мережеві конфігурації для підтримки тестування продуктивності.
4. Виконання тесту продуктивності: Сценарії тестування продуктивності будуть виконуватися в середовищі тестування продуктивності з використанням спеціалізованих інструментів тестування продуктивності. Тестувальники вимірюватимуть і контролюватимуть ключові показники продуктивності, такі як час відгуку, пропускна здатність і використання ресурсів, щоб оцінити продуктивність системи за різних умов навантаження.
5. Виконання стрес-тесту: Сценарії стрес-тесту будуть виконані для оцінки поведінки та стабільності програми в умовах екстремального навантаження. Ці сценарії можуть включати поступове збільшення робочого навантаження, введення сплесків користувацького трафіку або імітацію вичерпання ресурсів для виявлення вузьких місць і точок відмови системи.
6. Аналіз продуктивності: Показники продуктивності, зібрані під час виконання тесту, будуть проаналізовані для виявлення вузьких місць, обмежень масштабованості та областей для оптимізації. Аналіз продуктивності може включати порівняння показників продуктивності з попередньо визначеними еталонами, проведення аналізу першопричин проблем з продуктивністю та надання рекомендацій щодо покращення.
7. Звітність та рекомендації: Буде створено комплексний звіт про результати тестування продуктивності, в якому будуть задокументовані результати тестування, висновки та рекомендації щодо оптимізації продуктивності системи. Цей звіт буде надано відповідним зацікавленим сторонам, включаючи команди розробників, менеджерів проектів та системних архітекторів, щоб сприяти прийняттю обґрунтованих рішень та плануванню дій.

Дотримуючись цієї методології тестування продуктивності та стрес тестування, команда тестувальників прагне виявити та вирішити будь-які проблеми з продуктивністю або обмеженнями масштабованості в додатку TaskFlow, забезпечуючи оптимальну продуктивність і надійність в умовах різного робочого навантаження.

* 1. **User Acceptance Testing**

**Definition:**

1. Бажана комплексність: Тестування прийнятності для користувача (UAT) має на меті всебічно підтвердити, що додаток TaskFlow відповідає вимогам і очікуванням користувачів, забезпечуючи зручність, доступність і відповідність бізнес-цілям.
2. Методи оцінки комплексності: Комплексність в UAT буде оцінюватися на основі охоплення сценаріїв користувачів, відгуків від кінцевих користувачів та відповідності попередньо визначеним критеріям прийнятності. Ці критерії забезпечать ретельне тестування всіх критично важливих функціональних можливостей і робочих процесів користувачів.
3. Додаткові критерії завершення: Критерії завершення UAT можуть включати рейтинги задоволеності користувачів, успішне завершення користувацьких сценаріїв та валідацію ключових бізнес-процесів. Відгуки кінцевих користувачів і зацікавлених сторін також будуть враховуватися при оцінці завершеності UAT.
4. Методи відстеження вимог: Історії користувачів, критерії прийнятності та бізнес-вимоги будуть простежуватися до конкретних тестових кейсів і сценаріїв UAT, щоб гарантувати, що тестування адекватно охоплює всі вимоги та очікування користувачів

**Participants:**

До UAT залучатимуться кінцеві користувачі або представники цільової групи користувачів, які виконуватимуть заздалегідь визначені тестові сценарії та надаватимуть відгуки про додаток TaskFlow. Крім того, члени команди тестування та зацікавлені сторони проекту також можуть брати участь у плануванні, виконанні та оцінці UAT.

**Methodology:**

1. Розробка тестових сценаріїв: Сценарії тестування UAT будуть розроблені таким чином, щоб імітувати реальну взаємодію користувачів і робочі процеси, охоплюючи критичні функціональні можливості та сценарії загального використання. Ці сценарії будуть базуватися на історіях користувачів, критеріях прийнятності та бізнес-вимогах, забезпечуючи відповідність очікуванням користувачів.
2. Підготовка тестових даних: Для підтримки виконання тестових сценаріїв UAT будуть підготовлені набори тестових даних, що представляють реалістичні користувацькі сценарії та патерни використання. Ці набори даних включатимуть як правильні, так і неправильні вхідні дані, щоб перевірити поведінку системи за різних умов.
3. Налаштування тестового середовища: Буде створено спеціальне середовище UAT, яке максимально точно відтворюватиме виробниче середовище. Це середовище включатиме необхідне обладнання, програмне забезпечення та конфігурації для підтримки діяльності UAT.
4. Виконання UAT: Кінцеві користувачі або представники користувачів виконують заздалегідь визначені сценарії тестування UAT в середовищі UAT, дотримуючись задокументованих процедур і сценаріїв тестування. Тестувальники нададуть зворотній зв'язок щодо зручності використання системи, її доступності та відповідності бізнес-вимогам.
5. Звітування про дефекти: Будь-які дефекти або проблеми, виявлені під час UAT, будуть зареєстровані в системі відстеження дефектів з детальним описом, кроками для відтворення та оцінкою серйозності. Дефектам буде визначено пріоритети та призначено команди розробників для вирішення, з відповідним відстеженням та перевіркою виправлень.
6. Збір відгуків: На додаток до звітів про дефекти, кінцеві користувачі надаватимуть відгуки про свій загальний досвід роботи з додатком TaskFlow, включаючи зручність використання, продуктивність і відповідність бізнесцілям. Відгуки будуть збиратися за допомогою опитувань, інтерв'ю та аналітики використання.
7. Оцінка та прийняття: Результати UAT, включаючи результати тестування, звіти про дефекти та відгуки користувачів, будуть оцінюватися за заздалегідь визначеними критеріями прийнятності. Після успішного завершення UAT додаток TaskFlow буде вважатися готовим до запуску у виробництво.

Дотримуючись цієї методології тестування користувацького сприйняття, команда тестувальників прагне переконатися, що додаток TaskFlow відповідає вимогам, очікуванням і бізнес-цілям користувачів, надаючи продукт, який є зручним, інтуїтивно зрозумілим і відповідає потребам користувачів.

* 1. **Batch Testi**

**Definition:**

1. Бажана комплексність: Пакетне тестування має на меті всебічну перевірку функцій пакетної обробки в додатку TaskFlow, таких як імпорт, експорт і заплановані завдання.
2. Методи оцінки комплексності: Повнота пакетного тестування буде оцінюватися на основі покриття робочих процесів пакетної обробки, включаючи вхідні дані, перетворення і генерацію вихідних даних. Показники тестового покриття вказуватимуть на ступінь, до якого були протестовані різні функціональні можливості пакетної обробки.
3. Додаткові критерії завершення: Критерії завершення пакетного тестування можуть включати успішне виконання пакетних завдань, перевірку цілісності даних, точність вихідних результатів і дотримання попередньо визначених правил і норм обробки пакетів.
4. Методи відстеження вимог: Вимоги до пакетної обробки, правила перетворення даних і вихідні специфікації будуть відстежуватися до конкретних тестових кейсів і сценаріїв, щоб гарантувати, що тестування адекватно охоплює всі вимоги до пакетної обробки.

**Participants:**

У пакетному тестуванні братимуть участь члени команди тестування за підтримки розробників, які знайомі з функціоналом пакетної обробки та правилами перетворення даних.

**Methodology:**

1. Розробка тестових сценаріїв: Сценарії пакетного тестування будуть розроблені таким чином, щоб охопити різні робочі процеси пакетної обробки, включаючи прийом, перетворення і генерацію вихідних даних. Ці сценарії ґрунтуватимуться на вимогах до пакетної обробки, правилах перетворення даних і специфікаціях вихідних даних.
2. Підготовка тестових даних: Будуть підготовлені набори тестових даних, що представляють типові сценарії пакетної обробки та граничні випадки, для підтримки виконання сценарію пакетного тестування. Ці набори даних включатимуть як правильні, так і неправильні вхідні дані для перевірки логіки пакетної обробки та цілісності даних.
3. Налаштування тестового середовища: Буде створено спеціальне середовище для серійного тестування, яке відтворюватиме виробниче середовище серійної обробки. Це середовище включатиме необхідну інфраструктуру, бази даних та інструменти пакетної обробки для підтримки діяльності з пакетного тестування.
4. Виконання пакетних завдань: Сценарії пакетного тестування будуть виконуватися в середовищі пакетного тестування за допомогою автоматизованих інструментів пакетної обробки або скриптів. Тестувальники стежитимуть за виконанням пакетних завдань, перевірятимуть логіку перетворення даних і точність вихідних результатів.
5. Перевірка цілісності даних: Тестувальники перевірять цілісність даних, оброблених пакетними завданнями, гарантуючи, що вхідні дані правильно отримані, перетворені та завантажені в цільову систему. Перевірка даних буде виконуватися для виявлення будь-яких розбіжностей або помилок в оброблених даних.
6. Перевірка вихідних даних: Вихідні результати, отримані в результаті виконання пакетних завдань, будуть порівнюватися з очікуваними результатами, щоб перевірити правильність і повноту логіки пакетної обробки. Тестувальники перевірять, що вихідні файли і звіти створені точно і містять очікувані елементи даних.
7. Тестування обробки помилок: Сценарії обробки помилок будуть протестовані для перевірки здатності програми обробляти винятки, помилки та аномалії даних, що виникають під час пакетної обробки. Тестові кейси імітуватимуть умови помилок, щоб оцінити надійність механізмів обробки помилок.
8. Звітність та документація: Буде сформовано комплексний звіт про серійне тестування, в якому будуть задокументовані результати тестування, висновки та будь-які дефекти або проблеми, виявлені під час тестування. Цей звіт буде надано відповідним зацікавленим сторонам для полегшення прийняття рішень та планування дій.

Дотримуючись цієї методології пакетного тестування, команда тестувальників прагне забезпечити надійність, точність і ефективність функцій пакетної обробки в додатку TaskFlow, надаючи продукт, який відповідає бізнесвимогам і регуляторним стандартам

* 1. **Automated Regression Testing**

**Definition:**

1. Бажана комплексність: Автоматизоване регресійне тестування має на меті всебічну перевірку того, що існуючі функціональні можливості залишаються незмінними після зміни коду або оновлення системи.
2. Методи оцінки повноти: Комплексність автоматизованого регресійного тестування буде оцінюватися на основі покриття регресійних тестових кейсів, включаючи функціональне покриття, покриття інтеграції та покриття сценаріїв. Метрики тестового покриття вказуватимуть на ступінь, до якого були протестовані різні функціональні можливості.
3. Додаткові критерії завершення: Критерії завершення автоматизованого регресійного тестування можуть включати відсутність дефектів регресії, успішне виконання наборів регресійних тестів і дотримання попередньо визначених стандартів якості та еталонів.
4. Методи відстеження вимог: Регресійні тестові кейси будуть відстежуватися до конкретних функціональних вимог, історій користувачів або сценаріїв використання, щоб переконатися, що тестування адекватно охоплює всі зачеплені функціональні можливості.

**Participants:**

Автоматизоване регресійне тестування в першу чергу залучатиме членів команди тестування, зокрема інженерів з автоматизації тестування, які відповідають за розробку та підтримку автоматизованих тестових скриптів.

**Methodology:**

* 1. Розробка набору регресійних тестів: Набори регресійних тестів будуть розроблені таким чином, щоб охопити критичні функціональні можливості та робочі процеси користувачів, які схильні до регресійних дефектів. Ці набори тестів включатимуть комбінацію позитивних і негативних тестових кейсів для перевірки поведінки системи за різних умов.
  2. Розробка сценаріїв автоматизації тестування: Інженери з автоматизації тестування розроблятимуть та підтримуватимуть автоматизовані тестові скрипти з використанням фреймворків для автоматизації тестування, таких як Selenium, Appium або Cypress. Ці скрипти автоматизують виконання регресійних тестових кейсів для різних компонентів та модулів додатку TaskFlow.
  3. Підготовка тестових даних: Для підтримки автоматизованого регресійного тестування будуть підготовлені набори тестових даних, що представляють типові сценарії використання та граничні випадки. Ці набори даних включатимуть як правильні, так і неправильні вхідні дані для перевірки поведінки системи за різних умов.
  4. Налаштування тестового середовища: Буде створено спеціальне середовище автоматизації тестування для підтримки автоматизованого регресійного тестування. Це середовище включатиме необхідну інфраструктуру, інструменти автоматизації тестування та конфігурації для виконання автоматизованих тестових скриптів.
  5. Виконання регресійних тестів: Автоматизовані сценарії регресійного тестування будуть виконуватися в середовищі автоматизації тестування як частина конвеєра безперервної інтеграції та доставки (CI/CD). Фреймворки автоматизації тестування сприятимуть паралельному виконанню тестових скриптів, що дозволить проводити ефективне регресійне тестування декількох компонентів і модулів.
  6. Аналіз дефектів та звітування: Будь-які дефекти регресії, виявлені під час автоматизованого регресійного тестування, будуть зареєстровані в системі відстеження дефектів з детальним описом, кроками для відтворення та оцінкою серйозності. Дефектам буде визначено пріоритети та призначено команди розробників для усунення, з відповідним відстеженням та перевіркою виправлень.
  7. Обслуговування регресійних тестів: Сценарії регресійних тестів будуть регулярно переглядатися і оновлюватися, щоб забезпечити відповідність системним вимогам і змінам. Інженери з автоматизації тестування відстежуватимуть зміни в додатку TaskFlow і відповідно оновлюватимуть сценарії регресійних тестів, щоб підтримувати покриття та ефективність тестів.
  8. Звітність та документація: Буде сформовано комплексний звіт про регресійне тестування, в якому будуть задокументовані результати тестування, висновки та будь-які дефекти або проблеми, виявлені під час тестування. Цей звіт буде надано відповідним зацікавленим сторонам для полегшення прийняття рішень і планування дій.

Дотримуючись цієї методології автоматизованого регресійного тестування, команда тестувальників прагне забезпечити стабільність, надійність і ремонтопридатність додатку TaskFlow у наступних версіях і оновленнях, надаючи продукт, який відповідає стандартам якості і очікуванням користувачів.

* 1. **Beta Testing**

**Definition:**

1. Бажана комплексність: Бета-тестування має на меті всебічний збір відгуків від обраної групи зовнішніх користувачів або бета-тестерів перед офіційним випуском програми TaskFlow.
2. Методи оцінки комплексності: Комплексність бета-тестування буде оцінюватися на основі охоплення сценаріїв бета-тестування, різноманітності бета-тестерів, а також якості та кількості зібраних відгуків. Показники охоплення бета-тесту покажуть, наскільки різні аспекти програми були протестовані бета-тестерами.
3. Додаткові критерії завершення: Критерії завершення бета-тестування можуть включати збір достатнього і дієвого зворотного зв'язку, перевірку критично важливих функціональних можливостей і виявлення будь-яких дефектів або проблем, які можуть вплинути на рішення про випуск.
4. Методи відстеження вимог: Сценарії бета-тестування та зворотній зв'язок будуть відстежуватися відповідно до конкретних історій користувачів, критеріїв прийнятності або бізнес-вимог, щоб гарантувати, що тестування адекватно охоплює всі вимоги та очікування користувачів.

**Participants:**

У бета-тестуванні братиме участь обрана група зовнішніх користувачів або бета-тестерів, які представляють цільову аудиторію додатку TaskFlow. Крім того, члени команди тестування і зацікавлені сторони проекту також можуть брати участь у плануванні, проведенні та оцінці бета-тестування.

**Methodology:**

1. Набір бета-тестерів: Бета-тестери будуть набиратися з цільової аудиторії програми TaskFlow, включаючи існуючих користувачів, потенційних клієнтів та інші зацікавлені сторони. Критерії відбору можуть включати демографічні дані, моделі використання та готовність надавати зворотній зв'язок.
2. Розробка сценаріїв бета-тестування: Сценарії бета-тестування будуть розроблені таким чином, щоб охопити типові взаємодії користувачів і робочі процеси, зосереджуючись на критично важливих функціональних можливостях і сценаріях загального використання. Ці сценарії будуть базуватися на історіях користувачів, критеріях прийнятності та бізнесвимогах, забезпечуючи відповідність очікуванням користувачів.
3. Налаштування середовища бета-тестування: Буде створено спеціальне середовище бета-тестування, щоб надати бета-тестерам доступ до попередніх версій програми TaskFlow. Це середовище включатиме необхідну інфраструктуру, програмне забезпечення та конфігурації для підтримки бета-тестування.
4. Виконання бета-тесту: Бета-тестери виконуватимуть заздалегідь визначені сценарії бета-тестування в середовищі бета-тестування, дотримуючись задокументованих тестових процедур і скриптів. Тестувальники надаватимуть відгуки про зручність використання системи, функціональність, продуктивність та загальний досвід користувачів.
5. Збір відгуків: Відгуки бета-тестерів збиратимуться за допомогою опитувань, інтерв'ю, форм зворотного зв'язку та аналітики використання. Бета-тестери також матимуть можливість повідомляти про дефекти або проблеми, що виникли під час тестування, надаючи детальний опис, кроки для відтворення та оцінку серйозності.
6. Звітування про дефекти та їх вирішення: Будь-які дефекти або проблеми, виявлені під час бета-тестування, будуть зареєстровані в системі відстеження дефектів, пріоритезовані та призначені командам розробників для вирішення. Дефекти будуть усуватися оперативно, щоб гарантувати, що критичні проблеми будуть вирішені до офіційного випуску додатку TaskFlow.
7. Оцінка та рішення про випуск: Результати бета-тестування, включаючи відгуки, звіти про дефекти та загальний рейтинг задоволеності, будуть оцінюватися за заздалегідь визначеними критеріями прийнятності. На основі результатів бета-тестування буде прийнято рішення про випуск додатку TaskFlow, враховуючи такі фактори, як якість зібраних відгуків, серйозність виявлених дефектів і готовність додатку до випуску у виробництво

Дотримуючись цієї методології бета-тестування, команда тестувальників має на меті зібрати цінну інформацію від реальних користувачів, перевірити зручність використання та функціональність додатку TaskFlow, а також забезпечити успішний запуск, який відповідає очікуванням користувачів та бізнес цілям

1. **HARDWARE REQUIREMENTS**

Для оптимальної роботи програми TaskFlow необхідно виконати певні вимоги до апаратного забезпечення. Ці вимоги охоплюють необхідні обчислювальні пристрої та мережеву інфраструктуру. Вимоги до апаратного забезпечення є наступними:

**Computers:**

1. Сумісність з операційними системами: Додаток TaskFlow розроблений для роботи на різних операційних системах, включаючи Windows, macOS і дистрибутиви Linux. Сумісність з цими основними операційними системами забезпечує доступність для широкого кола користувачів.
2. Апаратні характеристики: Хоча TaskFlow може працювати на стандартних настільних і портативних комп'ютерах, рекомендовані технічні характеристики обладнання забезпечують оптимальну продуктивність. Для безперебійної роботи програми рекомендується мінімум 4 ГБ оперативної пам'яті. Крім того, двоядерний процесор забезпечує достатню обчислювальну потужність для ефективного виконання завдань. Також необхідний достатній обсяг пам'яті для розміщення самої програми та даних користувача.
3. Сумісність з браузерами: Сумісність з популярними веб-браузерами, такими як Google Chrome, Mozilla Firefox, Safari та Microsoft Edge, забезпечує доступність та зручність використання на різних комп'ютерних пристроях.

**Modems:**

1. Підключення до мережі: Якщо програма TaskFlow передбачає зв'язок із зовнішніми системами або віддалений доступ, модеми або мережеві пристрої необхідні для встановлення і підтримки мережевого з'єднання. Ці пристрої полегшують доступ до Інтернету, дозволяючи користувачам безперешкодно взаємодіяти з хмарними сервісами, отримувати доступ до віддалених серверів або використовувати функції онлайн-співпраці.
2. Типи модемів: TaskFlow підтримує різні типи модемів для роботи в різних мережевих середовищах. Дротові з'єднання Ethernet забезпечують стабільне і високошвидкісне з'єднання, що підходить для офісного середовища. Wi-Fi модеми забезпечують гнучкість і зручність бездротового доступу як вдома, так і в офісі. Крім того, мобільні широкосмугові модеми забезпечують підключення у віддалених або мобільних ситуаціях, коли дротові або Wi-Fi з'єднання можуть бути недоступні.

Враховуючи ці детальні вимоги до обладнання, користувачі можуть переконатися, що їхні комп'ютерні пристрої та мережева інфраструктура сумісні з додатком TaskFlow, тим самим оптимізуючи його продуктивність і функціональність відповідно до своїх конкретних потреб і сценаріїв використання

1. **ENVIRONMENT REQUIREMENTS**
   1. **Main Frame**

Технічні характеристики:

1. Фізичні характеристики: Середовище мейнфрейму для тестування TaskFlow повинно мати відповідні апаратні можливості для підтримки навантаження і масштабу програми. Це включає в себе високопродуктивні сервери з достатньою обчислювальною потужністю, пам'яттю і ємністю для зберігання даних, щоб впоратися з одночасною взаємодією користувачів і завданнями обробки даних.
2. Комунікаційне та системне програмне забезпечення: Середовище мейнфреймів має бути оснащене надійним комунікаційним та системним програмним забезпеченням, включаючи мережеві протоколи, системи управління базами даних та операційні системи, сумісні з вимогами програми TaskFlow.
3. Режим використання: Середовище мейнфреймів може функціонувати як централізована серверна інфраструктура, доступна для декількох користувачів з метою тестування. Воно може працювати в архітектурі клієнт-сервер, забезпечуючи віддалений доступ для команд тестувальників до взаємодії з додатком TaskFlow.
4. Вимоги до безпеки: Необхідно підтримувати високий рівень безпеки для захисту власних компонентів, таких як програмне забезпечення, дані та обладнання. Це передбачає впровадження контролю доступу, механізмів шифрування та регулярний аудит безпеки для захисту від несанкціонованого доступу, витоку даних та кіберзагроз.

Спеціальні тестові інструменти:

1. Для полегшення моніторингу продуктивності, навантажувального тестування та системної діагностики в середовищі мейнфреймів можуть знадобитися спеціалізовані інструменти тестування. Сюди входять інструменти моніторингу продуктивності, такі як New Relic або Dynatrace, інструменти тестування навантаження, такі як Apache JMeter або LoadRunner, а також інструменти системної діагностики для усунення несправностей і налагодження.
2. Крім того, для оцінки та зменшення ризиків безпеки в середовищі мейнфреймів можуть знадобитися інструменти тестування безпеки, такі як сканери вразливостей, фреймворки для тестування на проникнення, а також системи управління інформацією та подіями безпеки (SIEM).

Інші потреби в тестуванні:

1. Документація та навчання: Для ознайомлення команд тестувальників з середовищем мейнфреймів, його конфігурацією та процедурами тестування може знадобитися вичерпна документація та навчальні матеріали. Сюди входять посібники користувача, навчальні модулі та онлайн-ресурси для підтримки тестування.
2. Офісні приміщення: Для ефективної співпраці та ефективного проведення тестування командам тестувальників може знадобитися достатній офісний простір, обладнаний робочими місцями, кімнатами для переговорів та інструментами для співпраці. Це включає доступ до офісного обладнання, такого як столи, стільці, дошки та проектори для забезпечення потреб тестування.

Джерело потреб:

* 1. Закупівля апаратного та програмного забезпечення: Апаратні та програмні компоненти для середовища мейнфреймів, можливо, доведеться закуповувати у зовнішніх постачальників або ІТ-постачальників. Це передбачає пошук надійних і сумісних апаратних серверів, мережевого обладнання та ліцензійних програмних пакетів, які відповідають специфікаціям і вимогам середовища тестування TaskFlow.
  2. Впровадження заходів безпеки: Заходи безпеки для середовища мейнфреймів можуть вимагати співпраці з експертами або консультантами з кібербезпеки для оцінки ризиків безпеки, розробки політики безпеки та впровадження відповідних засобів контролю безпеки для захисту конфіденційних даних і ресурсів від несанкціонованого доступу та кіберзагроз.
  3. **Workstation**
  4. Робочі станції для тестування TaskFlow повинні відповідати мінімальним вимогам до апаратного та програмного забезпечення, визначеним для кінцевих користувачів. Це включає в себе настільні або портативні комп'ютери з достатньою обчислювальною потужністю, пам'яттю і ємністю для зберігання даних, щоб запустити додаток TaskFlow і ефективно виконувати тестування.
  5. Робочі станції повинні бути обладнані підтримуваними операційними системами та веб-браузерами, сумісними з TaskFlow, що забезпечить безперешкодний доступ до функцій та можливостей програми.
  6. Доступ до високошвидкісного інтернет-з'єднання необхідний для полегшення зв'язку з середовищем мейнфреймів, доступу до хмарних сервісів і спільної роботи в режимі онлайн під час тестування.

Забезпечивши відповідність середовищ мейнфреймів і робочих станцій зазначеним вимогам, команди тестувальників можуть провести ретельне тестування програми TaskFlow, виявити потенційні проблеми або вузькі місця в продуктивності, а також забезпечити готовність програми до розгортання у виробничих середовищах

1. **SCHEDULE**
2. Віхи випробувань:
   1. .Початок альфа-тестування: Починається фаза альфа-тестування, зосереджена на модульному тестуванні та початковому інтеграційному тестуванні. (Тривалість: 2 тижні) 1.2.Етап тестування системної інтеграції (SIT): Комплексне тестування системної інтеграції для перевірки взаємодії між різними компонентами. (Тривалість: 4 тижні)
   2. Етап тестування продуктивності: Тестування продуктивності та стрес-тестування для оцінки швидкості відгуку та масштабованості додатку. (Тривалість: 3 тижні)
   3. Фаза тестування на прийнятність для користувача (UAT): Бетатестування із зовнішніми користувачами для перевірки зручності та функціональності. (Тривалість: 2 тижні)
   4. регресійного тестування: Автоматизоване регресійне тестування для забезпечення стабільності існуючих функціональних можливостей. (Тривалість: постійно, інтегровано в конвеєр CI/CD)
   5. Фінальне приймальне тестування: Остаточна перевірка всіх вимог і функцій перед запуском у виробництво. (Тривалість: 1 тиждень)
3. Події передачі елементів:
   1. Надання тестового середовища: Налаштування середовища мейнфрейму та робочих станцій для тестування. (Заплановано до початку альфа-тестування)
   2. Випуск тестових скриптів: Розподіл тестових скриптів та сценаріїв командам тестувальників для виконання. (Заплановано до початку альфа-тестування)
   3. Розгортання тестових інструментів: Встановлення та налаштування спеціалізованих інструментів тестування для тестування продуктивності, безпеки та регресійного тестування. (Заплановано перед відповідними етапами тестування)
   4. Додаткові етапи тестування:
   5. Підготовка тестового середовища: Закупівля та налаштування обладнання, програмного забезпечення та мережевої інфраструктури для тестування. (Тривалість: 2 тижні)
   6. Навчання та адаптація: Тренінги для команд тестувальників для ознайомлення з процедурами, інструментами та середовищами тестування. (Тривалість: 1 тиждень)
   7. Наради з сортування дефектів: Регулярні зустрічі для визначення пріоритетів та призначення дефектів, виявлених під час тестування, для вирішення. (Заплановані щотижня на всіх етапах тестування)
   8. Документація та звітність: Підготовка тестових звітів, журналів дефектів та іншої документації для зацікавлених сторін. (Триває протягом усіх етапів тестування)
4. Розклад тестів:
   1. Старт альфа-тестування:
   2. Тривалість: 2 тижні (дата початку: 05.05.2024, дата закінчення: 18.05.2024)
   3. Ресурси: Команда тестувальників, тестове середовище 3.4.Віхи:
   4. Розпочато альфа-тестування, виявлено початкові дефекти
5. Етап тестування системної інтеграції (SIT):
   1. Тривалість: 4 тижні (дата початку: 05.05.2024, дата закінчення: 27.05.24)
   2. Ресурси: Команда тестувальників, середовище мейнфреймів, інструменти для тестування
   3. Етапи: Завершення SIT, усунення дефектів інтеграції
6. 5. Етап тестування продуктивності:
   1. Тривалість: 3 тижні (дата початку: 05.05.2024, дата закінчення: 21.05.2024)
   2. Ресурси: Команда тестувальників, інструменти для тестування продуктивності, середовище мейнфреймів
   3. Етапи: Встановлено контрольні показники продуктивності, протестовано масштабованість
7. Етап тестування користувацького сприйняття (UAT):
   1. Тривалість: 2 тижні (дата початку: 05.05.2024, дата закінчення: 18.05.2024)
   2. Ресурси: Зовнішні бета-тестери, тестове середовище, команда тестувальників
   3. 6.3.Етапи: Зібрано відгуки користувачів, перевірено юзабіліті
8. Етап регресійного тестування:
   1. Тривалість: Триває (інтегрований у конвеєр CI/CD)
   2. Ресурси: Автоматизовані інструменти тестування, команда тестувальників
   3. Етапи: Безперервне регресійне тестування, моніторинг тенденцій дефектів
9. Фінальне приймальне тестування:
   1. Тривалість: 1 тиждень (дата початку: 05.05.2024, дата закінчення: 12.05.2024))
   2. Ресурси: Команда тестувальників, тестове середовище
   3. .Етапи: Остаточна валідація всіх вимог, готовність до виробничого випуску

Дотримуючись цього розкладу та етапів тестування, можна ефективно керувати процесом тестування додатку TaskFlow, забезпечуючи ретельну перевірку всіх функціональних можливостей і готовність до промислового розгортання

1. **CONTROL PROCEDURES**

Повідомлення про проблеми

При виникненні інциденту в процесі тестування слід дотримуватися наступних процедур:

1. Ідентифікація інцидентів: Тестувальники повинні негайно ідентифікувати та документувати будь-які проблеми, дефекти або відхилення від норми, що виникають під час тестування. Це включає в себе фіксацію відповідних деталей, таких як кроки для відтворення проблеми, очікувана поведінка в порівнянні з фактичною, а також скріншоти або журнали, якщо вони доступні.
2. Реєстрація інцидентів: Інциденти слід реєструвати у визначеній системі відстеження інцидентів або репозиторії. Якщо використовується стандартна форма для звітування про інциденти, тестувальники повинні заповнити її всією необхідною інформацією, пов'язаною з інцидентом.
3. Оцінка серйозності та пріоритетності: Тестувальники або призначені зацікавлені сторони повинні оцінити серйозність і пріоритет кожного зареєстрованого інциденту на основі заздалегідь визначених критеріїв. Серйозність відноситься до впливу інциденту на систему або користувачів, тоді як пріоритетність вказує на терміновість вирішення проблеми.
4. Призначення та відстеження: Після реєстрації інциденти повинні бути призначені відповідним особам або командам, відповідальним за їх вирішення. Тестувальники повинні відстежувати статус кожного інциденту, щоб забезпечити його своєчасне вирішення та закриття.
5. Вирішення та верифікація: Розробники або призначені команди повинні своєчасно розслідувати та вирішувати повідомлені інциденти. Після вирішення інциденту тестувальники повинні перевірити виправлення, щоб переконатися, що проблема була вирішена належним чином.
6. Комунікація та документація: Протягом усього процесу вирішення інцидентів слід підтримувати чітку і своєчасну комунікацію між тестувальниками, розробниками, менеджерами проектів та іншими зацікавленими сторонами. Документування деталей інциденту, кроків вирішення та результатів перевірки повинно зберігатися для подальшого використання та аудиту.

Запити на зміну

Процес обробки змін у програмному забезпеченні, включаючи запити на зміну, виглядає наступним чином:

1. Ідентифікація змін: Будь-які запропоновані зміни до програмного забезпечення мають бути ідентифіковані та задокументовані як запити на зміну. Це можуть бути виправлення помилок, вдосконалення, нові функції або інші зміни, яких вимагають зацікавлені сторони.
2. Оцінка змін: Запити на зміни слід оцінювати на основі попередньо визначених критеріїв, включаючи аналіз впливу, оцінку доцільності, наявність ресурсів та відповідність цілям і вимогам проекту.
3. Пріоритетність змін: Запити на зміни повинні бути пріоритезовані на основі їхньої терміновості, важливості та потенційного впливу на графік і результати проекту. Рівні пріоритетів повинні бути призначені відповідно, щоб керувати послідовністю впровадження.
4. Затвердження змін: Зміни повинні пройти формальну перевірку та процес затвердження за участю відповідних зацікавлених сторін, включаючи спонсорів проекту, власників продукту, керівників розробки та представників відділу забезпечення якості. Лише затверджені зміни можна впроваджувати.
5. Впровадження змін: Затверджені зміни повинні бути впроваджені командою розробників відповідно до встановлених процедур кодування, тестування та розгортання. Зміни, що впливають на існуючі програми/модулі, повинні бути ідентифіковані та застосовані відповідні заходи контролю версій.
6. Перевірка змін: Після впровадження зміни слід ретельно протестувати, щоб перевірити їхню коректність, функціональність і сумісність з існуючим програмним забезпеченням. Слід провести регресійне тестування, щоб переконатися, що не виникне жодних небажаних побічних ефектів або регресії.
7. Документація змін: Документація щодо впроваджених змін, включно з журналами змін, примітками до релізу та оновленою користувацькою документацією, повинна бути підготовлена та передана відповідним зацікавленим сторонам. Це забезпечує прозорість та відстежуваність змін протягом усього життєвого циклу розробки програмного забезпечення.

Дотримуючись цих процедур контролю звітів про проблеми і запитів на зміни, команди тестування і розробки можуть ефективно управляти інцидентами, вносити зміни і забезпечувати якість і стабільність програми TaskFlow протягом усього її життєвого циклу.

1. **FEATURES TO BE TESTED**

Наступні програмні функції та комбінації програмних функцій будуть протестовані в додатку TaskFlow:

1. Аутентифікація та авторизація користувачів:
   1. Реєстрація користувача та вхід в систему
   2. Керування паролем (скидання, зміна)
   3. Контроль доступу на основі ролей
2. Управління завданнями:
   1. Створення завдань
   2. Призначення завдань користувачам
   3. Визначення пріоритетів та категоризація завдань
   4. Управління дедлайнами
   5. Відстеження статусу завдання (на розгляді, в процесі, виконано)
   6. Коментарі до завдань та вкладення
3. Управління робочим процесом:
   1. Створення та налаштування робочих процесів
   2. Призначення завдань в рамках робочих процесів
   3. Автоматизація робочого процесу та тригери
   4. Процес затвердження робочого процесу
4. Особливості співпраці:
   1. Спільна робота над завданнями в режимі реального часу
   2. Сповіщення та оповіщення
   3. Стрічки активності та оновлення
   4. Обмін файлами та контроль версій
5. Звітність та аналітика:
   1. Формування звітів про виконання завдань
   2. Аналітична панель для моніторингу продуктивності
   3. Налаштовувані шаблони звітів
   4. Експорт звітів у різні формати (PDF, CSV)
6. Інтеграція:
   1. Інтеграція зі сторонніми інструментами та сервісами (наприклад, календарем, електронною поштою)
   2. Інтеграція API для обміну даними
   3. Синхронізація даних із зовнішніми системами
7. Сумісність з мобільними пристроями:
   1. Адаптивний дизайн для мобільних пристроїв
   2. Власні функції мобільного додатку (якщо це можливо)
8. Функції безпеки:
   1. Шифрування даних під час передачі та в стані спокою
   2. Безпечні механізми автентифікації
   3. Контроль доступу на основі ролей
   4. Журнали аудиту та відстеження активності
9. Продуктивність і масштабованість:
   1. Навантажувальне тестування для оцінки продуктивності додатків під час пікових навантажень
   2. Тестування масштабованості для оцінки здатності системи справлятися зі збільшенням кількості користувачів і навантаження на дані
10. Юзабіліті та доступність:
    1. Інтуїтивно зрозумілий дизайн інтерфейсу
    2. Функції доступності для користувачів з обмеженими можливостями
    3. Юзабіліті-тестування для забезпечення задоволеності користувачів та простоти використання
11. Сумісність:
    1. Тестування кросбраузерної сумісності
    2. Сумісність з різними операційними системами та пристроями
12. Локалізація та інтернаціоналізація:
    1. Підтримка декількох мов і локалізацій
    2. Тестування мовних перекладів та культурної адаптації
13. **Resources/Roles & Responsibilities**

У тестовому проекті беруть участь наступні співробітники, кожен з яких виконує свої функції та обов'язки:

1. Керівник випробувань:
   1. Відповідає за загальне планування, координацію та управління тестуванням.
   2. Контролює весь процес тестування та забезпечує дотримання плану та графіку тестування.
   3. Керує ресурсами, бюджетом та звітністю.
2. Випробувальний свинець:
   1. Діє як основна контактна особа з питань тестування.
   2. Координує роботу з тестування, призначає завдання та контролює їх виконання.
   3. Співпрацює із зацікавленими сторонами для визначення пріоритетів тестування.
3. Аналітики/інженери з тестування:
   1. Розробляйте тестові плани, тестові кейси та тестові скрипти на основі вимог.
   2. Виконуйте тестові кейси та повідомляйте про дефекти.
   3. .Виконуйте функціональне, регресійне та продуктивне тестування.
4. Інженер з автоматизації:
   1. Розробляє та підтримує автоматизовані тестові скрипти для регресійного тестування.
   2. Впроваджує та керує фреймворками та інструментами автоматизації тестування.
   3. Співпрацює з тестовими аналітиками для визначення можливостей автоматизації тестування.
5. Аналітик з питань безпеки:
   1. Проводить тестування безпеки та оцінку вразливостей.
   2. Визначає ризики для безпеки та рекомендує стратегії їх зменшення.
   3. Співпрацює з розробниками для усунення вразливостей безпеки.
6. Команда тестування користувацького сприйняття (UAT):
   1. Представляє інтереси кінцевих користувачів під час тестування.
   2. .Перевіряє, чи відповідає система бізнес-вимогам та очікуванням користувачів.
   3. Забезпечує зворотній зв'язок щодо зручності та функціональності.
7. Команда розробників:
   1. Надає підтримку для аналізу та усунення дефектів.
   2. Впроваджує зміни та виправлення, виявлені під час тестування. \
   3. Забезпечує якість та цілісність коду.
8. Оперативний штаб:
   1. Керує налаштуванням та обслуговуванням тестового середовища.
   2. Налаштовує сервери, бази даних та мережеву інфраструктуру.
   3. Надає технічну підтримку в проведенні тестування.
9. Керівник проекту:
   1. Контролює загальний хід проекту, його бюджет і терміни.
   2. Співпрацює з тест-менеджером, щоб забезпечити відповідність діяльності з тестування цілям проекту.
   3. Керує комунікацією зі стейкхолдерами та їхніми очікуваннями.
10. Команда забезпечення якості (QA):
    1. Забезпечує нагляд та управління процесами тестування.
    2. Перевіряє результати тестування та забезпечує відповідність стандартам якості.
    3. Проводить аудит та визначає шляхи вдосконалення процесів.
11. Координатор тренувань:
    1. Розробляє та проводить навчальні програми для членів команди тестувальників.
    2. Забезпечує належне навчання членів команди інструментам, процесам та методологіям тестування.
    3. Надає постійну підтримку та рекомендації для покращення навичок тестування.
12. Постачальник послуг з тестування (за необхідності):
    1. Надає спеціалізовані послуги з тестування, такі як тестування продуктивності, безпеки або локалізації.
    2. Співпрацює з внутрішніми командами тестувальників, щоб розширити можливості та досвід тестування.
13. **Schedules**

Основні результати

Під час проекту тестування будуть створені наступні документи:

1. План випробувань:
   1. Очікувана доставка: 05.04.2024
   2. Опис: Комплексний документ, що описує підхід до тестування, стратегію, обсяг і ресурси для проекту тестування додатків TaskFlow.
2. 2. Тестові кейси:
   1. Очікувана доставка: 10..04..2024
   2. Опис: Детальні тестові кейси, що охоплюють різні сценарії та функції програми TaskFlow, включаючи позитивні та негативні тестові кейси.
3. Звіти про тестові інциденти:
   1. Очікувана поставка: Триває (по мірі виникнення інцидентів)
   2. Опис: Звіти, що документують інциденти, дефекти або аномалії, виявлені під час тестування, включаючи докладні описи, кроки для відтворення та оцінку серйозності.
4. Звіти про результати тестування:
   1. Очікувана поставка: В кінці кожного етапу тестування
   2. Опис: Зведені звіти, що містять огляд діяльності з тестування, включаючи покриття тестів, результати виконання, тенденції дефектів та рекомендації щодо подальших дій.

Розклад:

1. План випробувань:
   1. Проект плану випробувань: [Дата].
   2. Розгляд та затвердження: [Дата]
   3. Остаточний план випробувань: [Дата]
2. Тестові кейси:
   1. Розробка тестових кейсів: [Дата початку] - [Дата завершення]
   2. Розгляд та затвердження: [Дата]
3. Звіти про тестові інциденти:
   1. Триває протягом усіх етапів тестування
4. Звіти про результати тестування:
   1. Підсумковий звіт по альфа-тестуванню: [Дата].
   2. Звіт про результати тестування системної інтеграції: [Дата].
   3. Звіт про результати тестування продуктивності: [Дата].
   4. Звіт про результати тестування користувацького сприйняття: [Дата].
   5. Звіт про регресійне тестування: [Дата].
   6. Звіт про остаточні приймально-здавальні випробування: [Дата].

Дотримуючись цього графіку, ми забезпечуємо своєчасну доставку необхідної документації з тестування та підтримуємо прозорість і підзвітність протягом усього проекту тестування додатку TaskFlow.

1. **Significantly Impacted Departments (SIDs)**

|  |  |  |
| --- | --- | --- |
| **Department/Business Area** | **Business Manager** | **Tester(s)** |
| Product Development | Adam Ivanush | SviatoslavHnativ |
| Product Maneger | Volodya Hinzula | SviatoslavHnativ |

Ці відділи та бізнес-сфери мають значний вплив на проект тестування додатків TaskFlow. Визначені бізнес-менеджери і тестувальники з кожного відділу відповідають за координацію зусиль з тестування, надання зворотного зв'язку і забезпечення відповідності додатку вимогам і очікуванням своїх сфер діяльності. Співпраця та комунікація між цими відділами та командами тестувальників є запорукою успіху проекту тестування та загальної якості програми TaskFlow.

1. **DEPENDENCIES**

Існують значні обмеження при тестуванні програми TaskFlow:

1. Доступність тестового товару:
   1. Доступність тестових середовищ, включаючи мейнфрейми та робочі станції, може вплинути на діяльність з тестування. Затримки в налаштуванні або доступі до тестових середовищ можуть вплинути на загальний графік тестування.
2. Тестування - доступність ресурсів:
   1. Доступність ресурсів для тестування, таких як тест-аналітики, інженери з автоматизації, аналітики безпеки та користувацькі приймальники, може вплинути на темпи та ефективність тестування. Обмежена доступність кваліфікованих ресурсів може призвести до виникнення вузьких місць у процесі тестування.
3. Дедлайни:
   1. Кінцеві терміни і етапи проекту, включаючи дати випуску і графіки поставок, можуть накладати часові обмеження на діяльність з тестування. Тестування повинно бути завершене у визначені терміни, щоб переконатися, що додаток TaskFlow відповідає цілям проекту і узгоджується з бізнес-вимогами.
4. Залежності від розвитку:
   1. Діяльність з тестування залежить від виконання завдань розробки та доступності збірок програмного забезпечення для тестування. Затримки або зміни в графіку розробки можуть вплинути на терміни та обсяг робіт з тестування.
5. Інтеграція з третіми сторонами:
   1. Залежність від сторонніх систем або сервісів для інтеграційного тестування може призвести до невизначеностей і залежностей, які не піддаються контролю з боку команди тестувальників. Зміни або збої в роботі сторонніх систем можуть вплинути на графік і результати тестування.
6. Дотримання нормативних вимог:
   1. Дотримання нормативних стандартів і галузевих вимог може накладати додаткові обмеження на діяльність з тестування. Тестування повинно гарантувати, що додаток TaskFlow відповідає відповідним нормам і стандартам, що додає складності та обсягу зусиль по тестуванню.
7. Обмеженість ресурсів:
   1. Обмежена доступність інструментів тестування, ліцензій або ресурсів інфраструктури може обмежувати діяльність з тестування. Необхідно виділяти достатні ресурси для ефективної та результативної підтримки зусиль з тестування.

Усунення цих залежностей і обмежень вимагає ефективного планування, координації та комунікації між зацікавленими сторонами, командами тестувальників і менеджерами проектів. Щоб мінімізувати вплив обмежень на діяльність з тестування та забезпечити успішну реалізацію програми TaskFlow, слід впроваджувати стратегії пом'якшення наслідків

1. **RISKS/ASSUMPTIONS**

Припущення з високим ступенем ризику:

1. Наявність тестових середовищ:
   1. Припущення: Тестові середовища будуть доступні відповідно до запланованого графіку.
   2. Ризик: Затримки в налаштуванні або доступі до тестових середовищ можуть вплинути на терміни та результати тестування.
   3. План дій у надзвичайних ситуаціях: Підтримуйте тісний зв'язок з командами ІТ-інфраструктури, щоб пришвидшити налаштування середовища. Якщо необхідно, вивчіть альтернативні варіанти тестування, такі як хмарні середовища.
2. Доступність ресурсів:
   1. .Припущення: Достатня кількість ресурсів для тестування, включаючи кваліфікованих тестувальників та інженерів з автоматизації, буде доступна протягом усього проекту тестування.
   2. Ризик: Обмеженість ресурсів тестування може призвести до виникнення вузьких місць у тестуванні та вплинути на загальну якість та покриття тестування.
   3. План на випадок непередбачених обставин: Проведіть перехресне навчання членів команди критичним завданням тестування, щоб пом'якшити ресурсні обмеження. Розгляньте можливість аутсорсингу певних видів діяльності з тестування зовнішнім постачальникам, якщо внутрішніх ресурсів недостатньо.
3. Залежність від третіх осіб:
   1. Припущення: Сторонні системи та сервіси, необхідні для інтеграційного тестування, будуть доступні та функціонувати як очікується.
   2. Ризик: Затримки або збої в роботі сторонніх систем можуть вплинути на тестування інтеграції та внести невизначеність у результати тестування.
   3. План дій у надзвичайних ситуаціях: Встановіть чіткі канали зв'язку зі сторонніми постачальниками для моніторингу доступності системи та оперативного вирішення будь-яких проблем. За необхідності розробіть альтернативні підходи до тестування або імітаційні сервіси для імітації взаємодії зі сторонніми постачальниками.
4. Відповідність та регуляторні вимоги:
   1. Припущення: Додаток TaskFlow відповідатиме всім відповідним регуляторним стандартам та галузевим вимогам.
   2. Ризик: невідповідність нормативним стандартам може призвести до юридичних або фінансових наслідків і вплинути на розгортання та використання програми.
   3. План дій у надзвичайних ситуаціях: Проводьте ретельні перевірки та аудити відповідності протягом усього процесу тестування. Залучайте юридичні та комплаєнс-групи для перевірки діяльності з тестування та забезпечення дотримання регуляторних вимог. Вживайте коригувальних заходів за необхідності для усунення прогалин у дотриманні вимог.
5. Управління змінами:
   1. Припущення: Зміни до програмного забезпечення будуть ефективно управлятися, а їхній вплив на діяльність з тестування буде адекватно вирішуватися.
   2. Ризик: погано керовані зміни можуть призвести до нестабільності, переробок і розширення обсягу, що вплине на графік і результати тестування.
   3. План дій у надзвичайних ситуаціях: Впровадьте надійні процеси управління змінами, щоб відстежувати та оцінювати зміни в програмному забезпеченні. Проводьте оцінку впливу, щоб визначити потенційні наслідки тестування та відповідно скоригувати плани та графіки тестування

Врахування цих припущень з високим рівнем ризику за допомогою планів дій у надзвичайних ситуаціях допоможе пом'якшити потенційний вплив на проект тестування і забезпечити успішні результати тестування програми TaskFlow. Регулярна оцінка ризиків і проактивні стратегії управління ризиками повинні застосовуватися протягом усього процесу тестування, щоб мінімізувати невизначеності і максимізувати ефективність тестування

1. TOOLS

Наступні інструменти автоматизації та інструмент відстеження помилок будуть використовуватися для проекту тестування додатків TaskFlow:

Інструменти автоматизації:

1. 1. Selenium WebDriver:
   1. Призначення: Selenium WebDriver буде використовуватися для автоматизованого функціонального тестування веб-додатку TaskFlow. Він дозволяє створювати надійні та підтримувані тестові скрипти для автоматизації роботи браузера.
2. 2. Jenkins:
   1. Призначення: Jenkins буде використовуватися як інструмент безперервної інтеграції та безперервної доставки (CI/CD) для автоматизації виконання тестових сценаріїв, побудови розгортань і генерації тестових звітів.
3. Apache JMeter:
   1. Мета: Apache JMeter буде використовуватися для тестування продуктивності та стрес-тестування додатку TaskFlow. Він дозволяє створювати тестові плани для моделювання різних сценаріїв навантаження та вимірювання продуктивності системи за різних умов.
4. Postman:
   1. Призначення: Postman буде використовуватися для тестування API, дозволяючи тестувальникам розробляти, тестувати і документувати API швидко і ефективно.

Інструмент відстеження помилок:

1. Jira:
   1. Призначення: Jira слугуватиме основним інструментом відстеження помилок та управління проблемами для проекту тестування TaskFlow. Він дозволяє тестувальникам повідомляти, відстежувати та визначати пріоритети дефектів, а також організовано керувати тестовими завданнями та діяльністю.

Використовуючи ці інструменти автоматизації та інструмент відстеження помилок, команда тестування може впорядкувати процеси тестування, підвищити ефективність і забезпечити ефективне управління дефектами і проблемами протягом усього життєвого циклу тестування додатку TaskFlow

1. APPROVALS

Цей план випробувань повинен бути затверджений наступними особами:

|  |  |  |
| --- | --- | --- |
| **Name (In Capital Letters)** | **Signature** | **Date** |
| Adam Ivanush |  |  |
| Volodya Hinzula |  |  |
| Dmutro Ryzhytskyi |  |  |
| Sviatoslav Hnativ |  |  |