

Завдання 1

Розписати roadmap (карту) зі всіма видами тестів та їх методами.

Roadmap (карту) зі всіма видами тестів та їх методами		
Назва тестування	Мета	Методи
Юніт-тестування (Unit Testing)	Перевірка окремих модулів або функцій у коді.	Параметризовані тести для перевірки різних вхідних значень. Тестування з використанням фреймворків.
Інтеграційне тестування (Integration Testing)	Перевірка взаємодії між компонентами або модулями.	Bottom-up (від нижчих до вищих модулів). Top-down (від вищих до нижчих модулів). Підходи "Big Bang" та інкрементальне тестування.
Системне тестування (System Testing)	Перевірка роботи всього продукту у реальному середовищі.	Функціональне тестування. Нефункціональне тестування (перевірка продуктивності, надійності тощо). Перевірка сумісності.
Приймальне тестування (Acceptance Testing)	Перевірка, чи відповідає продукт бізнес-вимогам.	Альфа-тестування (внутрішнє тестування замовником). Бета-тестування (тестування на реальних користувачах).
Тестування продуктивності (Performance Testing)	Оцінка швидкості, стабільності та продуктивності системи.	Навантажувальне тестування. Стрес-тестування. Тестування стабільності.
Тестування безпеки (Security Testing)	Виявлення вразливостей та захист від зловмисних атак.	Тестування проникнення. Аудит коду. Тестування на відповідність стандартам.
Регресійне тестування (Regression Testing)	Перевірка, що нові зміни не вплинули на існуючий функціонал.	Повне регресійне тестування. Вибіркове регресійне тестування.
Тестування юзабіліті (Usability Testing)	Оцінка зручності використання продукту для кінцевих користувачів.	Спостереження. А/В тестування. Анкетування та інтерв'ю.

Завдання 2 Подумайте та наведіть приклади, які види тестів можна використовувати на різних рівнях тестування.

		Види тестів:
1. Модульне тестування (Unit Testing)	Цей рівень тестування спрямований на перевірку окремих компонентів або модулів системи.	Юніт-тести: Перевіряють функції, класи, методи окремо. Статичне тестування коду: Аналіз без виконання програми (linting, статичні аналізатори коду). Мок-тестування: Використання заглушок (mock objects) для ізоляції модуля. Boundary Testing: Перевірка граничних умов.
2. Інтеграційне тестування (Integration Testing)	На цьому рівні перевіряється взаємодія між модулями.	Тести інтеграції компонентів: Перевірка зв'язків між двома або більше модулями. API-тести: Перевіряють інтеграцію через інтерфейси. Тести контрактів: Гарантують, що взаємодія між модулями відповідає специфікаціям. Тести сумісності: Перевіряють, чи правильно інтегруються компоненти різних систем.
3. Системне тестування (System Testing)	На цьому рівні перевіряється вся система як єдине ціле.	Функціональне тестування: Перевірка, чи відповідає система вимогам. Нефункціональне тестування: Тестування продуктивності (Performance Testing): Перевірка швидкості системи. Тестування безпеки (Security Testing): Виявлення вразливостей. Тестування навантаження (Load Testing): Випробування системи під великим навантаженням. Тестування доступності (Accessibility Testing): Перевірка доступності для людей з обмеженими можливостями. Регресійне тестування: Перевірка, чи не з'явилися нові помилки після змін у системі.
4. Приймальне тестування (Acceptance Testing)	На цьому рівні перевіряється, чи відповідає система очікуванням замовника або кінцевих користувачів.	Тестування за сценаріями (User Acceptance Testing): Імітація дій користувачів. Альфа-тестування: Проводиться в середовищі розробника з участю користувачів. Бета-тестування: Тестування продукту реальними користувачами у їхньому середовищі. Тести на відповідність вимогам: Перевірка, чи відповідає система стандартам.
Додаткові види тестів, які можуть застосовуватись на будь-якому рівні: Димове тестування (Smoke Testing): Перевірка базової працездатності системи. Санітайз тестування (Sanity Testing): Швидка перевірка після внесення змін. Тестування сценаріїв (End-to-End Testing): Випробування повного потоку роботи системи.		

Завдання 3

Наведіть порядок тестування, пов'язаного зі зміною, якщо до вас потрапляє новий функціонал та виправлений дефект.

Оцінка зміни	Оцінка функціональної зміни	Оцінити, яку саме частину програми було змінено. Якщо це новий функціонал, з'ясувати, як він взаємодіє з іншими частинами системи. Якщо виправлений дефект, визначити, як саме зміна вплине на продукт і чи можуть виникнути нові дефекти.
	Оцінка впливу	Визначити, які модулі або частини системи можуть бути порушені через новий функціонал або виправлений дефект.
Розробка тестових випадків	Тестування нового функціоналу	Створити нові тестові випадки для перевірки доданого функціоналу. Вони повинні охоплювати всі сценарії використання нової функції.
	Тестування виправлення дефекту	Перевірити, чи дефект, який був виправлений, не з'явився знову, а також чи не виникли нові проблеми після його виправлення.
	Регресійне тестування	Оскільки зміни можуть впливати на інші частини програми, проведення регресійного тестування необхідне для перевірки, чи не порушено існуючий функціонал.
Тестування нових змін	Функціональне тестування	Перевірити, чи новий функціонал працює відповідно до вимог. Інтеграційне тестування: Переконавшись, що новий функціонал правильно інтегрується в систему і не порушує її взаємодію з іншими компонентами.
	Тестування на продуктивність та безпеку	Якщо зміна може вплинути на продуктивність або безпеку, провести відповідні тести.
Виконання регресійного тестування	Перевірка існуючих функцій	Протестувати старі функції, які могли зазнати змін. Це дозволить виявити можливі проблеми, викликані новим функціоналом чи виправленням дефекту.
	Автоматизовані тести	Якщо є автоматизовані тестові скрипти, їх також варто виконати для забезпечення швидкої перевірки всіх основних функцій.
Перевірка якості документації	Оновлення документації	Якщо новий функціонал або виправлення дефекту змінюють існуючі процедури або архітектуру, оновити відповідні документи (технічні, користувацькі інструкції тощо)
Ретестинг	Перевірка дефектів	Якщо після виправлення дефекту існують певні умови, які потрібно перевірити повторно (особливо після виправлення), то необхідно провести ретестинг цих випадків.
Завершення тестування	Збір і аналіз результатів	Перевірити всі результати тестування та скласти звіт. За потреби, уточнити або доповнити тестові випадки для подальших перевірок.
	Передача результатів	Якщо всі тести успішно пройшли, передати продукт або зміни на наступну стадію розробки або релізу.