**Лекція 19 Мобільне тестування**

**Мобільне тестування - це процес перевірки та оцінки якості мобільних додатків і пристроїв перед їхнім випуском на ринок**. Він включає в себе серію тестів, які виконуються на різних рівнях, щоб переконатися, що додаток функціонує належним чином і відповідає очікуванням користувачів.

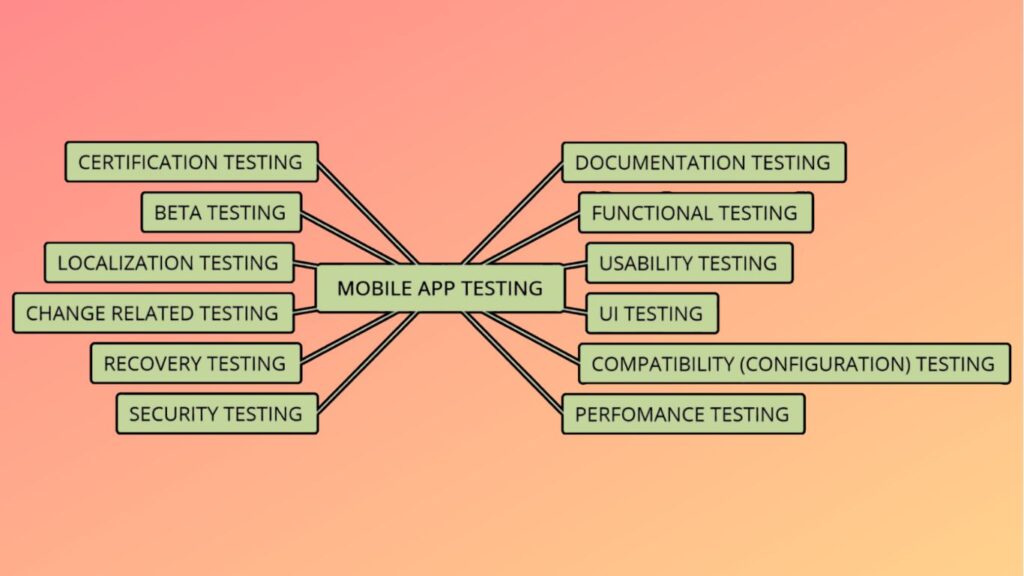
Мобільне тестування (МТ) стало практично невід’ємною частиною нашого сучасного життя. Воно відіграє ключову роль у забезпеченні високого рівня продуктивності та функціональності додатків, а також безпеки наших даних.

**Що таке мобільне тестування?**

Це процес перевірки та оцінки якості мобільних додатків і пристроїв перед їхнім випуском на ринок. Він включає в себе серію тестів, які виконуються на різних рівнях, щоб переконатися, що додаток функціонує належним чином і відповідає очікуванням користувачів.

Існують такі основні типи тестування:

* Функціональне (працездатність застосунку та відповідність функціональним вимогам).
* Зовнішніх подій (робота застосунку в умовах дзвінків, смс, відсутності мережі та сповіщень).
* Безпеки (наявності вразливостей і стійкості до злому, захист від несанкціонованого доступу до інформації).
* Встановлення (коректності встановлення програми, особливо попередніх версій).
* Продуктивності (перевірка серверної частини застосунку, особливо в разі використання специфічних SDK *-* Software Development Kit).
* Локалізації (робота застосунку в обраній країні, переклади текстів та інтерфейсів).
* Юзабіліті (зручність використання, враховуючи обмеження екрана, фрагментацію пристроїв і роботу однією рукою).



Тестування мобільних додатків – це процес, за допомогою якого прикладне ПЗ, розроблене для портативних мобільних пристроїв, перевіряється на його функціональність, зручність використання та сумісність. Тестування може бути мануальним або автоматизованим.

1. Функціональне тестування є найбільш базовим тестом на перевірку відповідності вимогам для будь-якої програми. Як і інші додатки, засновані на призначеному для користувача інтерфейсі, мобільні додатки вимагають ряду взаємодій людини у сценаріях користувача.
2. Тестування сумісності найважливіше тоді, коли справа доходить до тестування мобільних додатків. Мета тестування мобільного додатку на сумісність, як правило, полягає в тому, щоб ключові функції програми працювали належним чином на конкретному пристрої. Сама сумісність повинна займати всього кілька хвилин і може бути заздалегідь спланованою. Прийняти рішення, які тести на сумісність мобільних пристроїв слід виконати, є нелегким завданням (оскільки тестування на усіх існуючих пристроях просто неможливе). Тому необхідно підготувати тестову матрицю з кожною можливою комбінацією та розставити пріоритети для клієнта.
3. Localization Testing. Сьогодні більшість додатків призначені для глобального застосування, і тому дуже важливо пам’ятати про регіональні особливості, такі як мови, часові пояси і т.д. Важливо перевірити функціональність програми, коли користувач змінює часовий пояс. Необхідно враховувати, що іноді західні дизайни можуть не працювати з аудиторією зі східних країн або навпаки.
4. Laboratory testing, зазвичай проводиться мережевими операторами, виконується моделюванням усієї бездротової мережі. Цей тест виконується з метою виявлення будь-яких збоїв у тому випадку, коли мобільний додаток використовує передачу голосу та/або дані про виконання окремих функцій.
5. Performance Testing охоплює продуктивність клієнтських додатків, сервера та мережі. Завдяки Performance Testing можна ідентифікувати існуючі мережі, сервери та вузькі місця серверних додатків, враховуючи визначене навантаження та поєднання транзакцій.
6. Stress Testing є обов’язковим тестуванням під час виявлення винятків, зависань та взаємних блокувань, що може залишитися непоміченими під час тестування функціональності й інтерфейсу користувача.

Ось список деяких критеріїв:

* Завантажте у свій додаток якомога більше даних, спробувавши досягти його межі.
* Виконуйте знову і знову ті ж самі операції.
* Виконуйте повторно операції на різних швидкостях, дуже швидко або дуже повільно.
* Залиште ваш додаток працювати протягом тривалого періоду часу, одночасно взаємодіючи з пристроєм та просто залишаючи його непрацюючим, або виконуючи певне автоматичне завдання, яка займає багато часу, наприклад, слайд-шоу.
* Випадково відправляти екранні натискання та натискання клавіш у вашому додатку.
* На вашому пристрої повинні бути запущені кілька додатків, щоб ви могли часто перемикатися між додатком та іншими додатками на пристрої.

1. Security Testing допомагає виявити всі можливі вразливості щодо політик злому, автентифікації й авторизації, безпеки даних, управління сеансами та інших стандартів безпеки. Додатки повинні шифрувати ім’я користувача і паролі під час автентифікації користувача у мережі.

Один із способів тестування сценаріїв, пов’язаних з безпекою, полягає у маршрутизації даних вашого мобільного пристрою через проксі-сервер,наприклад такий, як OWASP Zed Attack Proxy, і пошуку уразливості.

1. Usability Testing оцінює додаток на основі наступних трьох критеріїв для цільової аудиторії: ефективність; точність і повнота; задоволеність. Дуже важливо провести юзабіліті-тестування на найбільш ранньому етапі розробки програми. Цей вид тестування вимагає активної участі користувачів, і результати можуть вплинути на дизайн програми, що дуже важко змінити на пізніших етапах проєкту.

Існує ще безліч тестів, які необхідно провести під час тестування мобільного додатка:

* Installation/Uninstallation testing
* Updates Testing
* Certification Testing
* Screen Orientation / Resolution
* Memory Leakage Testing
* Available Tools
* Touch Screens
* Soft & Hard Keys

**Важливість мобільного тестування**

Для користувачів вигідно, коли застосунок працює стабільно, без збоїв і помилок, це підвищує їхній рівень задоволеності та впевненість у надійності програми. З точки ж зору бізнесу, якість застосунку відіграє вирішальну роль у його успіху. Такий застосунок приваблює більше користувачів, утримує їхню увагу та підвищує їхню лояльність. Він сприяє позитивній репутації бренду, покращує імідж компанії, що веде до зростання довіри та залучення нових клієнтів.

Особливості тестування мобільних додатків полягають у врахуванні специфічних аспектів і вимог, пов’язаних із мобільними пристроями та їхнім оточенням. Ось деякі з них:

* різноманітність пристроїв і платформ;
* адаптація до різних роздільних здатностей екранів;
* тестування за різних умов мережі;
* обробка зовнішніх подій і переривань (дзвінки, повідомлення, сповіщення та інші зовнішні події, які можуть вплинути на роботу програми);
* зручний користувацький інтерфейс для сенсорних екранів;
* перевірка мобільних функцій і можливостей.

Таким чином, тестування застосунків сприяє поліпшенню користувацького досвіду, підвищенню задоволеності користувачів і успішності мобільного застосунку загалом.

**Процес мобільного тестування**

Тестування додатків на мобільних пристроях в цілому відповідає загальним принципам тестування. Але також присутній ряд особливостей, які характерні саме для тестування мобільних додатків.

Для того щоб зрозуміти особливості тестування додатків на мобільних пристроях, необхідно враховувати чинники, які відрізняють мобільний додаток від десктопного, а саме: специфічні й різноманітні операційні системи для мобільних платформ, різні конфігурації комплектуючих, функціональність таких пристроїв, як комунікатори та ін.

У зв'язку з цими факторами, підхід до тестування додатків на мобільних пристроях досить сильно відрізняється від десктопного. З'являється велика кількість додаткових нюансів та вимог, які необхідно протестувати.

Давайте розглянемо основні моменти, на які необхідно звертати особливу увагу саме при тестуванні додатків на мобільних пристроях.

**Розмір екрана та touch-інтерфейс:**

* розміри всіх елементів графічного інтерфейсу користувача;
* перевірка можливості використання усіх активних елементів (кнопок, посилань та ін.);
* швидкість відгуку активних елементів повинна бути досить високою;
* перевірка того, що багаторазове швидке натискання на кнопку не викличе екстрене завершення програми;
* підтримка мультитач – одночасне натискання кількох кнопок;
* підтримка горизонтального (landscape) та вертикального (portrait) положення;
* перевірка використання в додатку спеціальних жестів (double tap, swipe, pinch in/out та ін.).

**Ресурси телефону:**

* необхідно проконтролювати можливі витоки пам'яті. Часто це трапляється в додатках з вікнами, що містять велику кількість інформації. Також витоки пам'яті можуть бути присутніми під час тривалої роботи програми;
* перевірка обробки ситуацій нестачі пам'яті для функціонування операційної системи, під час роботи програми в активному та фоновому режимах;
* недолік місця для установки або роботи програми;
* установка на SD-карту;
* перевірка роботи батареї (акумулятора) пристрою при запущеному додатку, роботи у фоновому режимі, при включеному Wi-Fi, 3G Інтернеті, без підключення до мережі та ін.

**Різні роздільні здатності екрану та версії ОС:**

* необхідно перевірити роботу програми на пристроях з різними роздільними здатностями екрану. На екранах з високою роздільною здатністю (наприклад, ретіна-екран) елементи інтерфейсу та текст відображаються дрібніше, при роботі додатка на пристрої з екраном більш низькою роздільною здатністю – елементи інтерфейсу можуть стати занадто великими;
* необхідно переконатися, що додаток не може бути встановлений на непідтримувані пристрої. При цьому обов'язково тестування програми на усіх заявлених підтримуваних пристроях.

**Реакція програми на зовнішні переривання:**

* вхідні та вихідні SMS та MMS;
* вхідні та вихідні дзвінки;
* нагадування, будильник та ін.;
* відключення та підключення Wi-Fi. Наприклад, додаток може екстрено завершувати свою роботу під час запуску авіарежиму при відключеному Wi-Fi. При втраті сигналу та одночасному виконанні операції може відображатися нескінченне завантаження програми, замість коректного повідомлення про відсутність інтернет-підключення;
* часто виникають проблеми з переходом в онлайн-режим після офлайн режиму;
* відключення та підключення SD-карти;
* відключення/підключення мобільного пристрою до зарядного пристрою;
* робота з фізичною клавіатурою (якщо в списку підтримуваних моделей є такі);
* робота в фоновому (background) режимі. Для відправки у фоновий режим – запускається додаток, а потім натискається кнопка home. В результаті при повторному запуску програми бувають помилки, екстрені завершення роботи, неправильне відображення останнього відкритого вікна. Також при тривалій бездіяльності додатка виникає екстрене завершення роботи або помилки;
* робота додатка після виходу зі сплячого режиму;
* сумісність з іншими додатками.

Користувач мобільного пристрою очікує, що додатки, які встановлюються, прості, інтуїтивно зрозумілі, працюють завжди й всюди без збоїв. Якщо очікування не виправдовуються, то користувач попросту видаляє додаток та встановлює аналогічний додаток від іншого розробника, яких у сфері мобільних розробок завжди достатньо. Тому якість програми є одним з головних факторів його популярності.

**Процес мобільного тестування** включає такі етапи:

1. Планування і підготовка:

* визначення цілей і вимог тестування;
* створення тестового плану, що містить опис тестових сценаріїв і ресурси;
* вибір пристроїв і операційних систем для цього процесу;
* встановлення необхідного ПЗ та інструментів.

1. Виконання тестових сценаріїв:

* запуск програми на обраних пристроях і ОС;
* виконання заздалегідь визначених тестових сценаріїв, які включають різні дії та перевірки;
* тестування різних функціональних аспектів додатка, таких як навігація, введення даних, обробка помилок тощо;
* відтворення та реєстрація помилок і несправностей.

1. Аналіз і документація результатів:

* аналіз отриманих результатів тестування;
* документування виявлених помилок, включно з їхнім описом, кроками для відтворення та оцінкою їхньої серйозності;
* взаємодія з розробниками для уточнення і виправлення помилок;
* підготовка звітів про виконане тестування, включно із загальною статистикою, знайденими помилками та рекомендаціями щодо поліпшення якості застосунку.

**Функціональне тестування мобільного додатку**

Це процес тестування функціональних можливостей мобільних програм, таких як взаємодія користувачів, а також тестування транзакцій, які користувачі можуть виконувати. Основною метою функціонального тестування мобільних додатків є забезпечення якості, відповідність заданим очікуванням, зниження ризику або помилок і задоволення клієнтів.

Різноманітні фактори, які мають значення для функціонального тестування, є

1. Тип програми на основі використання бізнес-функцій (банківська, ігрова, соціальна чи бізнес)
2. Тип цільової аудиторії (споживач, підприємство, освіта)
3. Канал розповсюдження, який використовується для поширення програми (наприклад, Apple App Store, Google play, пряме розповсюдження)

Найбільш фундаментальними тестовими сценаріями у функціональному тестуванні можна вважати:

1. Щоб перевірити, чи всі необхідні обов’язкові поля працюють належним чином.
2. Щоб переконатися, що обов’язкові поля відображаються на екрані в інший спосіб, ніж необов’язкові поля.
3. Щоб перевірити, чи програма працює відповідно до вимог кожного разу, коли програма запускається/зупиняється.
4. Щоб перевірити, чи програма переходить у згорнутий режим щоразу, коли надходить вхідний телефонний дзвінок. Щоб підтвердити те саме, нам потрібно використати другий телефон, щоб зателефонувати на пристрій.
5. Щоб перевірити, чи здатний телефон зберігати, обробляти та отримувати SMS, коли програма запущена. Щоб перевірити те саме, нам потрібно використовувати другий телефон, щоб надіслати SMS на пристрій, який тестується і на якому зараз запущена тестована програма.
6. Щоб підтвердити, що пристрій здатний виконувати необхідні багатозадачні вимоги, коли це необхідно.
7. Щоб підтвердити, що програма підтримує необхідні параметри соціальних мереж, такі як обмін, публікація та навігація тощо.
8. Щоб підтвердити, що програма підтримує будь-які транзакції платіжного шлюзу, такі як Visa, Mastercard, Paypal тощо, як того вимагає програма.
9. Щоб перевірити, чи сценарії прокручування сторінки ввімкнено в програмі за необхідності.
10. Щоб підтвердити, що навігація між відповідними модулями в додатку відповідає вимогам.
11. Щоб підтвердити, що помилки скорочення є абсолютно доступними.
12. Щоб переконатися, що користувач отримує відповідне повідомлення про помилку, наприклад «Помилка мережі. Будь ласка, спробуйте через деякий час», коли виникає будь-яка помилка мережі.
13. Щоб переконатися, що встановлена програма забезпечує задовільну роботу інших програм і не поглинає пам’ять інших програм.
14. Щоб підтвердити, що програма відновлюється в разі жорсткого перезавантаження або збою системи.
15. Щоб перевірити, чи можна встановити програму без проблем, за умови, що користувач має необхідні ресурси, і це не призводить до значних помилок.
16. Щоб підтвердити, що програма виконує автоматичний запуск відповідно до вимог.
17. Щоб перевірити, чи програма працює відповідно до вимог у всіх версіях Mobile, тобто 2g, 3g і 4g.
18. Виконувати Регресійне тестування щоб виявити нові помилки програмного забезпечення в існуючих областях системи після внесення в них змін. Також повторно запустіть раніше виконані тести, щоб визначити, що поведінка програми не змінилася через зміни.
19. Щоб перевірити, чи програма надає доступний посібник користувача для тих, хто не знайомий із програмою

**Тестові приклади тестування продуктивності**

Основна мета цього типу тестування полягає в тому, щоб забезпечити прийнятну роботу програми за певних вимог до продуктивності, таких як доступ великої кількості користувачів або видалення ключової частини інфраструктури, наприклад сервера бази даних.

Загальні сценарії тестування продуктивності в мобільній програмі:

1. Щоб визначити, чи програма працює відповідно до вимог за різних умов навантаження.
2. Щоб визначити, чи здатне поточне покриття мережі підтримувати програму на піковому, середньому та мінімальному рівнях користувачів.
3. Щоб визначити, чи забезпечує існуюча конфігурація клієнт-сервер необхідний оптимальний рівень продуктивності.
4. Для виявлення різних вузьких місць програми та інфраструктури, які перешкоджають роботі програми на необхідних рівнях прийнятності.
5. Щоб перевірити, чи час відповіді заявки відповідає вимогам.
6. Щоб оцінити продукт та/або апаратне забезпечення, щоб визначити, чи може воно витримувати прогнозовані обсяги навантаження.
7. Щоб оцінити, чи може час роботи батареї підтримувати роботу програми за прогнозованих обсягів навантаження.
8. Для перевірки продуктивності програми, коли мережу змінено на WIFI з 2G/3G або навпаки.
9. Для перевірки кожного необхідного циклу ЦП проводиться оптимізація
10. Щоб підтвердити, що споживання батареї, витоки пам’яті, такі ресурси, як GPS, продуктивність камери відповідають вимогам.
11. Щоб перевірити довговічність програми, коли навантаження користувача є жорстким.
12. Для перевірки продуктивності мережі під час переміщення з пристроєм.
13. Для перевірки продуктивності програми, коли потрібні лише періодичні фази підключення.

## Тестові випадки тестування безпеки

Основна мета тестування безпеки полягає в тому, щоб переконатися, що вимоги безпеки даних і мереж до програми відповідають інструкціям.

Наступні є найбільш важливими областями для перевірки безпеки мобільних додатків.

1. Щоб підтвердити, що програма здатна протистояти будь-якій атаці грубої сили, яка є автоматизованим процесом проб і помилок, який використовується для вгадування імені користувача, пароля або номера кредитної картки.
2. Щоб перевірити, чи програма не дозволяє зловмиснику отримати доступ до конфіденційного вмісту або функцій без належної автентифікації.
3. Щоб підтвердити, що програма має astrong система захисту паролем і не дозволяє зловмиснику отримати, змінити або відновити пароль іншого користувача.
4. Щоб переконатися, що програма не страждає від недостатнього закінчення сеансу.
5. Визначити динамічні залежності та вжити заходів, щоб запобігти доступу будь-якого зловмисника до цих вразливостей.
6. Для запобігання від SQL атаки, пов'язані з ін'єкціями.
7. Для виявлення та відновлення будь-яких сценаріїв некерованого коду.
8. Щоб переконатися, що сертифікати перевірені, чи реалізовано програмою закріплення сертифіката чи ні.
9. Для захисту програми та мережі від атак типу «відмова в обслуговуванні».
10. Проаналізувати вимоги до зберігання та перевірки даних.
11. Щоб увімкнути керування сеансом для запобігання доступу неавторизованих користувачів до небажаної інформації.
12. Щоб перевірити, чи не зламано будь-який криптографічний код, і переконатися, що його відремонтовано.
13. Щоб перевірити, чи реалізація бізнес-логіки захищена та чи не вразлива до будь-яких атак ззовні.
14. Щоб проаналізувати взаємодію файлової системи, визначити будь-яку вразливість і виправити ці проблеми.
15. Для перевірки обробників протоколів, наприклад, спроба змінити налаштування цільової сторінки за замовчуванням для програми за допомогою шкідливого iframe.
16. Для захисту від зловмисних ін’єкцій на стороні клієнта.
17. Для захисту від шкідливих ін’єкцій під час виконання.
18. Щоб дослідити кешування файлів і запобігти будь-яким шкідливим можливостям з нього.
19. Щоб запобігти незахищеному зберіганню даних у кеші клавіатури програм.
20. Щоб перевірити файли cookie та запобігти будь-яким зловмисним діям із файлів cookie.
21. Проводити регулярні аудити для аналізу захисту даних.
22. Досліджуйте створені на замовлення файли та запобігайте будь-яким зловмисним діям із створених на замовлення файлів.
23. Для запобігання від buffer випадки переповнення та пошкодження пам'яті.
24. Для аналізу різних потоків даних і запобігання будь-якій уразливості в них.

## Тестування юзабіліті

Процес перевірки зручності використання мобільного додатка виконується, щоб отримати швидкий і простий додаток із меншою функціональністю, ніж повільний і складний додаток із багатьма функціями. Основна мета полягає в тому, щоб ми мали простий у використанні, інтуїтивно зрозумілий і схожий на загальноприйнятий у галузі інтерфейс, який широко використовується.

1. Для цього кнопки повинні мати необхідний розмір і підходити для великих пальців.
2. Переконайтеся, що кнопки розташовані в одній частині екрана, щоб уникнути плутанини для кінцевих користувачів.
3. Щоб переконатися, що піктограми є природними та відповідають програмі.
4. Щоб переконатися, що кнопки, які виконують однакові функції, також повинні мати однаковий колір.
5. Щоб переконатися, що перевірка для натискання zoom-в і zoom-вихідні засоби повинні бути ввімкнені.
6. Щоб переконатися, що введення з клавіатури можна мінімізувати відповідним чином.
7. Щоб переконатися, що програма надає метод для повернення або скасування дії, торкаючись неправильного елемента, протягом прийнятного періоду часу.
8. Щоб переконатися, що контекстне меню не перевантажене, оскільки його потрібно використовувати швидко.
9. Для того, щоб текст був простим і зрозумілим для видимості користувачам.
10. Щоб переконатися, що короткі речення та абзаци є readable кінцевим користувачам.
11. Щоб переконатися, що розмір шрифту достатньо великий readable і не занадто великий чи занадто малий.
12. Щоразу, коли користувач починає завантажувати великий обсяг даних, що може не сприяти продуктивності програми, користувач запитує перевірку програми.
13. Щоб підтвердити, що закриття програми виконується з різних станів, і перевірити, чи вона знову відкривається в тому самому стані.
14. Щоб переконатися, що всі рядки перетворюються на відповідні мови, коли доступна функція перекладу мови.
15. Щоб переконатися, що елементи програми завжди syncхронізуються відповідно до дій користувача.
16. Щоб гарантувати, що кінцевому користувачеві надається посібник користувача, який допомагає кінцевому користувачеві зрозуміти.

Тестування зручності використання зазвичай виконується користувачами, оскільки лише люди можуть зрозуміти чутливість і комфорт інших користувачів.

## Тестові випадки тестування сумісності

Тестування сумісності на мобільних пристроях виконується для того, щоб переконатися, що мобільні пристрої мають різні розміри, роздільну здатність, екран, версію та апаратне забезпечення, тому програму слід тестувати на всіх пристроях, щоб переконатися, що програма працює належним чином.

Наступні є найбільш помітними областями для перевірки сумісності.

1. Щоб підтвердити, що інтерфейс користувача програми відповідає розміру екрана пристрою, текст/елемент керування не є частково невидимим або недоступним.
2. Для того, щоб текст був читабельним для всіх користувачів програми.
3. Щоб переконатися, що функція виклику/будильника ввімкнена під час роботи програми. Програма згортається або призупиняється під час виклику, а потім, коли виклик припиняється, програма відновлюється.

## Тестування відновлюваності

1. Відновлення після збою та переривання транзакцій
2. Перевірка ситуації ефективного відновлення програми після неочікуваних сценаріїв переривання/збою.
3. Перевірка того, як програма обробляє транзакцію під час збою живлення (наприклад, батарея розряджається або раптове ручне вимкнення пристрою)
4. Перевірку процесу, у якому з’єднання призупинено, системі необхідно відновити для відновлення даних, на які безпосередньо вплинуло призупинене з’єднання. Користуючись правом засоби мобільного тестування може допомогти забезпечити безперебійний процес відновлення.

### Важливий контрольний список

1. Тестування інсталяції (чи можна інсталювати програму за розумний проміжок часу та за необхідним критерієм)
2. Перевірка деінсталяції (чи програму можна видалити за розумний проміжок часу та за необхідним критерієм)
3. Випадки тестування мережі (перевірка того, чи працює мережа під необхідним навантаженням чи ні, чи здатна мережа підтримувати всі необхідні програми під час процедур тестування)
4. Перевірте невідповідні ключі
5. Перевірте заставку програми
6. Продовження введення з клавіатури під час переривань та інших випадків, наприклад проблем з мережею
7. Методи, які мають справу з виходом із програми
8. Ефект зарядного пристрою, коли програма працює у фоновому режимі
9. Низький заряд батареї та висока продуктивність
10. Виймання батареї під час виконання програми
11. Споживання батареї за програмою
12. Перевірте побічні ефекти програми

# Тестування переривань у мобільному додатку

Тестування переривань — це розділ тестування мобільних додатків, який має справу з тим, як програма реагує на переривання та повертається до попереднього стану. Тестування переривань намагається з’ясувати, яку поведінку демонструє ваша програма.

Залиште на секунду всі фрази вбік і подивіться на іншу реальну ситуацію. Припустімо, у вас є ліхтарик і ви ввімкнули його. Акумулятор розряджається, що є перервою в його поточному стані активності. Замініть батареї та відновіть його. Ліхтарик має знову засвітитися, як зазвичай. Це варіант використання. Дисципліна тестування, яка зосереджена на тому, відбувається це чи ні, — це тестування переривань.

Тестування переривань застосовується до будь-яких типів додатків – веб, мобільних, автономних тощо. Різноманітність пристроїв, мереж, конфігурацій тощо робить його більш помітним для Mobile програм, ніж інші.

## Тип перерв у мобільному додатку

* Низький заряд батареї
* Акумулятор повний - під час зарядки
* Вхідний телефонний дзвінок
* Вхідні SMS
* Вхідне сповіщення з іншої мобільної програми
* Підключено для зарядки
* Відключено від зарядки
* Пристрій вимкнено
* Нагадування про оновлення програми
* сигнал тривоги
* Втрата з'єднання з мережею
* Відновлення підключення до мережі

Цей список не є вичерпним, але містить найпоширеніші сценарії.

## Роздільна здатність у разі переривання

Очікувана поведінка у випадку цих переривань є одним із наступних:

1. **Запуск у фоновому режимі:** Переривання бере на себе, тоді як програма відходить на другий план. Він отримує контроль після закінчення перерви. Наприклад, телефонний дзвінок/Facetime який ви відвідуєте, коли читаєте цифрову книгу в iBooks (або подібній програмі). Коли користувач відповідає на телефонний дзвінок, iBooks чекає, поки це буде зроблено, а потім відновлює розмову після завершення розмови.
2. **Показати сповіщення. Сповіщення зникає, і ви працюєте як зазвичай. '**SMS отримано'- повідомлення з'являються в заголовку. Користувач не турбується про це і продовжує працювати з додатком у звичайному режимі. Інші сповіщення мобільних додатків, як-от новий запит у друзі на Facebook або WhatsApp повідомлення, також потрапляють у цю категорію. Але якщо користувач вирішить прочитати повідомлення, діятиме, як описано в пункті 1. Якщо ігнорувати, стан програми не змінюється.
3. **Заклик до дії**: будильники потрібно вимкнути або відкласти, перш ніж продовжити роботу. Те ж саме з повідомленнями про оновлення програми. Вам потрібно скасувати або прийняти зміни, перш ніж продовжити. Іншим прикладом є попередження про низький заряд батареї. Ви можете продовжити, як зазвичай, або перейти в режим низького енергоспоживання (якщо пристрій це дозволяє).
4. **Немає впливу:** Приклад: якщо підключення до мережі стає доступним і ваш пристрій підключається до нього. Крім того, коли ви підключаєте свій пристрій для заряджання, жодних сповіщень або закликів до дії не потрібно. Ймовірно, він виконає свою роботу, поки ви продовжуєте використовувати програму.

Таким чином, залежно від переривання, яке ви тестуєте, зрозумійте поведінку та подивіться, чи ваша програма задовольняє її. Крім того, поведінка, описана вище, не повинна бути однаковою для всіх програм і пристроїв. Обов'язково з'ясуйте конкретні деталі про ваш мобільний додаток.

## Як виконати тестування переривань

Тестування переривань є підмножиною Функціонального тестування для мобільного додатку. І щоб провести тестування переривань, ви повинні використовувати ті самі рамки та інструменти тестування мобільних додатків. Створити ці сценарії — це майстерність тестувальників. Після цього ви розробите тестові випадки та виконайте їх точно так само, як і будь-який інший тест.

**Інструменти та технології**

На ринку існує безліч інструментів і технологій, які полегшують процес мобільного тестування. Перелічимо деякі з популярних:

* **Appium**. Безкоштовний і відкритий інструмент для автоматизації мобільного тестування, який підтримує додатки для iOS, Android і Windows. Дозволяє писати тести різними мовами програмування, включно з Java, Python і JavaScript.
* **Espresso**. Інструмент для автоматизації функціонального тестування Android-додатків. Пропонує простий API для написання тестів і дає змогу швидко виконувати тести на пристроях та емуляторах.
* **XCUITest**. Фреймворк для автоматизації мобільного тестування додатків для iOS. Забезпечує широкі можливості для тестування інтерфейсу, продуктивності та функціональності додатків на пристроях Apple.
* **Calabash**. Інструмент для функціонального тестування мобільних додатків на платформах iOS і Android. Дозволяє писати тести мовою Ruby і забезпечує можливість взаємодії з інтерфейсом додатків.
* **Selendroid**. Фреймворк для автоматизації тестування Android-додатків. Підтримує різні мови програмування, включно з Java і C#, і забезпечує можливість виконання тестів на реальних пристроях і емуляторах.

*Всі згадані інструменти надають різноманітні можливості та функціональності, завдяки яким автоматизація тестування мобільних застосунків стає легшою та швидшою.*

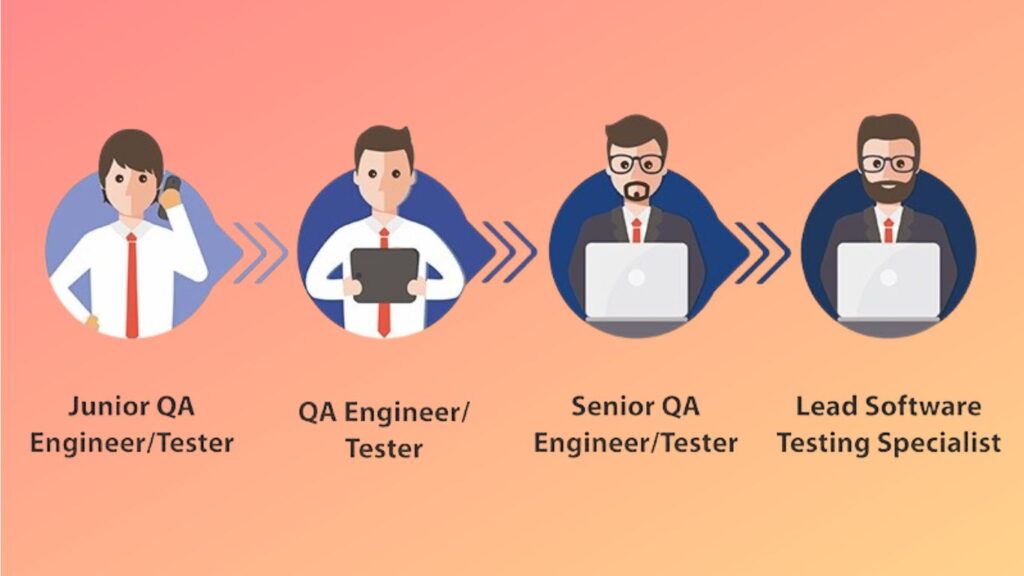
Цей підхід – дедалі популярніший в індустрії розробки мобільних застосунків, адже має безліч переваг для процесу тестування. Наприклад:

* підвищення швидкості;
* збільшення покриття;
* поліпшення точності та надійності;
* економія часу та ресурсів;
* виявлення дефектів на ранніх стадіях.

**Навичками і знаннями тестувальника мобільних застосунків**

Він повинен володіти певними навичками і знаннями. Одне з ключових умінь – це планування і проектування тестових сценаріїв. Це включає здатність аналізувати вимоги до застосунку, виділяти тестові випадки і створювати плани тестування. А ще, тестувальник мобільних додатків:

* уважний до деталей;
* має гарну пам’ять;
* володіє аналітичним мисленням;
* має схильність до точних наук;
* цілеспрямований і націлений на результати;
* здатний працювати в малорухомих умовах;
* знає основні методики та техніки тестування мобільних додатків;
* має досвід роботи з популярними інструментами тестування, такими як Appium, Espresso, XCUITest та ін.;
* розуміє основні ОС і платформи мобільних пристроїв;
* уміє документувати результати тестування.



**Переваги та недоліки**

Мобільне тестування надає низку переваг, які роблять його невід’ємною частиною розробки мобільних додатків:

* Широке охоплення користувачів. Мобільні пристрої мають величезну базу користувачів, і тестування на них дає змогу досягти широкої аудиторії.
* Реалістичне середовище. МТ проводиться в реальних умовах використання, що допомагає виявити проблеми, пов’язані з мережею, мобільними ОС і апаратними обмеженнями.
* Різноманітність платформ. Мобільні пристрої працюють на різних операційних системах, як-от iOS і Android, що вимагає тестування на різних платформах.
* Чуйність і взаємодія. Оскільки відбувається процес взаємодії з користувачем через сенсорні екрани, жести та інші вступні методи, тестування допомагає перевірити чуйність програми.

Однак існують і деякі труднощі, з якими стикаються фахівці з МТ:

* Фрагментація пристроїв. Мобільний ринок різноманітний за моделями, роздільною здатністю екрана, версіями операційних систем, що ускладнює тестування. Рішення: використання апаратних ферм тестування та емуляторів для покриття різних конфігурацій.
* Швидке оновлення ОС. Подібне відбувається досить часто, і додатки мають бути сумісними з новими версіями. Рішення: регулярне оновлення тестових оточень і адаптація тестів до нових версій ОС.
* Різноманітність сенсорів і функцій. Мобільні пристрої мають безліч сенсорів і функцій, які потребують тестування, таких як GPS, камера, акселерометр та інші. Рішення: розробка специфічних тестових сценаріїв для кожної функції та використання відповідних інструментів.

Крім цього, важливим є постійне навчання і самоосвіта в галузі мобільних технологій та інструментів тестування, поліпшення комунікації та співпраця з іншими учасниками команди розроблення, тестування на реальних пристроях, постійний аналіз і поліпшення процесу тестування;

**Висновок**

Мобільне тестування відіграє ключову роль у розробці якісних мобільних додатків. Воно допомагає виявити помилки, підвищити задоволеність користувачів і поліпшити репутацію компанії. Незважаючи на деякі складнощі, пов’язані зі швидким розвитком технологій, впоратися з ними можна завдяки постійному навчанню, автоматизації тестування, поліпшенню комунікації та аналізу процесу.

**Питання для самоконтролю:**

1.   Чому мобільне тестування важливе?.

2.   У чому ж особливості тестування мобільних додатків?

3.   Які вимоги до фахівця із тестування мобільних додатків?

4.   Що таке тестування переривань?

5. Навіщо тестування переривань?

6. Як виконати тестування переривань?