

## Дослідження методів тестування додатка для мобільних пристроїв

Чешун К.А., Орленко В.С., Джулій В.М.  
Хмельницький національний університет

Процес розробки додатків для мобільних пристроїв являє собою ітеративну схему, її застосування доцільно в проектах розробки додатка для мобільних пристроїв, загальною тривалістю до одного року. Особливість ітеративної схеми полягає в прискореному просуванні до готового продукту внаслідок кількох ітерацій повного циклу розробки (рис. 1).

В результаті визначення функціональних вимог починається цикл з декількох етапів розробки готових альфа-версій продукту. На кожному етапі відбувається повна збірка додатка, системне тестування нового функціоналу і етап регресійного тестування. Після кількох ітерацій приймається рішення, що подальше доопрацювання додатка не є доцільним і відбувається вихід з циклу з додатком, готовим до випуску. Процес визначення вимог вимагає попереднього аналізу цільової аудиторії і ринку аналогічної продукції. Можливі кілька варіантів визначення бізнес вимог до продукту: «мозковий штурм» за участю всіх залучених до процесу розробки людей; зустрічі з замовником і формування високорівневих вимог. Основна особливість отриманих вимог - високий рівень абстракції. Вимоги повинні бути зрозумілі будь-якій людині, залученій в процес розробки, і не повинні залежати від конкретної технології розробки. Існує кілька способів подання і зберігання бізнесі вимог:

- набір шаблонів - паперових або електронних. Кожен шаблон описує певну частину програми. Шаблони статичні і ніяк не взаємодіють між собою. Очевидні плюси шаблонів - їх швидка розробка і наочність. Шаблони можна розробити таким чином, щоб вони дуже точно описували призначений для користувача інтерфейс продукту, що розробляється;

- модель додатка, написана на формальній мові. Для розробки формальних моделей необхідне знання відповідних інструментів моделювання, наприклад, UniTESK. Подібні формальні моделі дозволяють точно описувати внутрішню структуру програми, наприклад, ООП класи. Також формальні моделі дозволяють генерувати функціональні специфікації та тестові сценарії на основі побудованих моделей. Тестові сценарії, що генеруються, можуть застосовуватися для тестування методом «білого ящика», оскільки формальна модель описує внутрішню структуру програми. Такі моделі не дозволяють «побачити» готовий додаток, оскільки описують тільки функціонал, але не призначений для користувача інтерфейс;

- модель додатка, створена за допомогою спеціальних інструментів прототипування інтерфейса для користувача. Прототипи дозволяють точно описати поведінку додатка на системному рівні, тобто інтерфейс для користувача. При цьому прототипи ніяк не описують внутрішню структуру програми. Прототипи дозволяють генерувати функціональні специфікації та

тестові сценарії, придатні для тестування методом «чорного ящика».

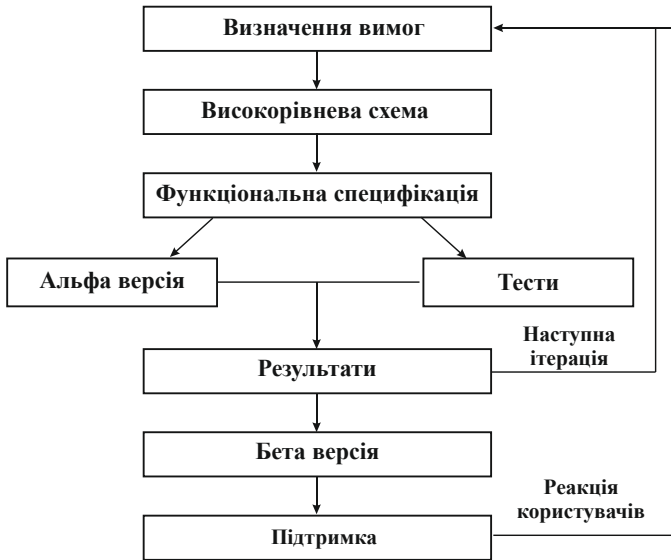


Рисунок 1 - Ітеративна схема розробки додатка для мобільних пристроїв

Проведений аналіз процесу розробки додатка для мобільних пристроїв дозволяє виділити наступні особливості: процес розробки розглянутих додатків швидкоплинний, загальна тривалість проектів по впровадженню програми - не більше одного року; розробка відбувається по ітеративній схемі, тому вимагає оптимізації інтегрованих процесів функціонального тестування за часовим параметром; розробка високорівневого прототипу додатка з подальшою генерацією тестових сценаріїв, імовірно, дозволить оптимізувати процес впровадження додатків.

В ітеративній схемі, в процесі розробки додатків для мобільних пристроїв, додаток надходить на тестування на системній стадії, минаючи процеси модульного і інтеграційного тестування. Таким чином, процес функціонального тестування зводиться до перевірки функціональності програми на рівні призначеного для користувача інтерфейсу. У загальному випадку взаємодія користувача з додатком відбувається за наступною схемою:

1. Користувач бачить на екрані мобільного пристрою деякий «вид» додатка. Цей вид містить елементи призначеного для користувача інтерфейсу (кнопки, поля введення і ін.), які дозволяють здійснювати різні дії (запити користувача).

2. Запит, що генерується користувачем, надходить в «логічну» частину

додатка, яка обробляє запит і, можливо, звертається до бази даних за необхідними даними.

3. Отримавши дані, «логічна» частина генерує наступний «вид», який побачить користувач - як результат свого запиту.

4. Далі процес повторюється.

В схемі, зображеній на рис. 2, на кожному етапі розробки можна вибрати один із способів переходу до наступного етапу. Залежно від обраного переходу змінюється час досягнення наступного етапу. Сумарний час розробки залежить від шляху проходження зі стану 1 схеми в стан 7 і від кількості ітерацій циклу 1-7. Зі стану 1 в стан 7 можна прийти різними шляхами, але в кожному стані вибір наступного маршруту залежить від попереднього стану процесу розробки. Кожен з можливих маршрутів визначає метод тестування програми. Розглянемо найбільш поширені шляхи переходу з 1 в 7.

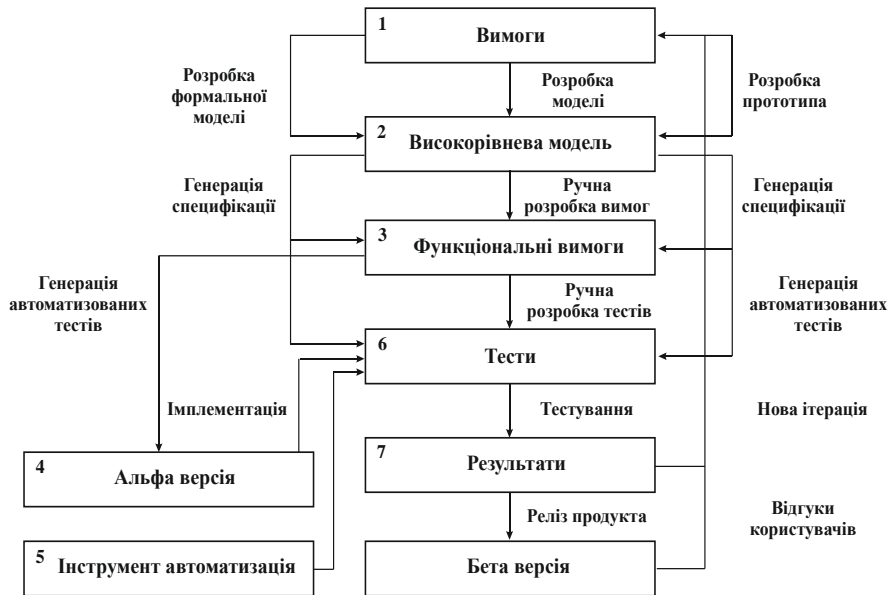


Рисунок 2 - Детальна ітеративна схема розробки додатка для мобільних пристроїв

Ручне тестування з документації: визначення вимог; розробка шаблонів; ручна розробка текстової функціональної специфікації; ручна розробка тестових сценаріїв; ручне тестування. Метод ручної розробки документів і тестів (PP метод) дуже гнучкий і дозволяє досягти будь-якого рівня тестового покриття. Однак цей метод вимагає великих часових витрат.

Автоматизація ручних тестів: визначення вимог; розробка шаблонів;

ручна розробка текстової функціональної специфікації; ручна розробка тестових сценаріїв в текстовому вигляді; автоматизація тестових сценаріїв; автоматизоване тестування.

Метод автоматичної розробки тестів (*РА* метод) вимагає текстового опису тестових сценаріїв. По суті, створюються тестові сценарії в текстовому вигляді, як і в випадку *PP* методу, а на основі даних тестових сценаріїв розробляються автоматизовані тести. Такий підхід вимагає значно більше часу на розробку тестів, але значно скорочує час проведення тестування. Отже, такий підхід ефективний при збільшенні кількості циклів розробки.

Тестування на основі формальної моделі (*ФМ* метод): визначення вимог; розробка формальної моделі; генерація специфікації; генерація тестових сценаріїв; автоматизоване тестування. Даний метод підлягає автоматизації, але не досить гнучкий в області розробки додатків для мобільних пристроїв, оскільки за допомогою формальних моделей важко описувати призначені для користувача інтерфейси, отже, важко отримати необхідне тестове покриття.

Тестування на основі прототипу: визначення вимог; розробка прототипу; генерація специфікації; генерація тестових сценаріїв; ручне та автоматизоване тестування. Метод розробки на основі прототипу (*ПП* метод) дозволяє добре описувати призначені для користувача інтерфейси, отже, отримати необхідне покриття. Метод також дозволяє генерувати різні типи тестових сценаріїв при наявності відповідних інструментів генерації і особливі правила побудови прототипів, що надає додаткову гнучкість в проведенні тестування.

Для мобільних додатків існують різні інструменти автоматизації тестування. У загальному випадку можна виділити два основні підходи до автоматизації тестів.

Програмний метод автоматизації. Метод, який передбачає використання певних бібліотек деякої мови програмування. Автоматизація тестів зводиться до написання допоміжних модулів тестування і написання тестових скриптів на мові програмування. Основний мінус цього підходу - значні часові витрати на реалізацію тестових скриптів.

Метод автоматизації з використанням *playback* інструментів. Такий підхід вимагає мінімальних навичок в програмуванні. Автоматизація тесту відбувається за допомогою проведення тестового сценарію на цільовому пристрої з пропущенням впливів через спеціальну програму (проксі-сервер), що записує вплив і дані. Після запису ця програма представляє отриману інформацію у вигляді автоматичного тесту. При такому підході автоматизація тесту зводиться до його виконання. Основний мінус цього підходу - будь-яка зміна в тестовому додатку вимагає перезапису тестових сценаріїв, пов'язаних зі зміненням функціоналом.

Виходячи з проведеного дослідження, можна зробити наступні висновки щодо методів тестування додатка для мобільних пристроїв: ефективність методу тестування додатка для мобільних пристроїв безпосередньо залежить від кількості ітерацій розробки, тому оптимізація методів тестування за часом зводиться до автоматизації процесів створення тестових сценаріїв і автоматизації їх виконання; найбільшу ефективність надає метод тестування на основі прототипів, при цьому тестові сценарії для тестування додатка для мобільних пристроїв генеруються з побудованого прототипу; необхідно мати можливість генерувати тестові сценарії таким чином, щоб використовувати їх в програмному методі автоматизації тестування з найменшими доопрацюваннями.

#### Література

1. Ханссон Д. Х., Томас, Д. Гибкая разработка веб-приложений в среде Rails / Д. Томас, Д. Х. Ханссон — Санкт-Петербург - Питер, 2008.
2. Салмре Иво Конечный автомат для пользовательского интерфейса. Программирование мобильных устройств на платформе .NET Compact Framework / Иво Салмре – Изд-во: Издательский дом "Вильямс", 2006.
3. Джулій В.М. Методи та алгоритми розробки web-додатків / В.М. Джулій, Ю.О. Гунченко, Д.В. Чешун // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – К.: ВІКНУ, 2017. – Вип. № 56. – С.107-115
4. Роберт Мартин Быстрая разработка программ. Принципы, примеры, практика. /Роберт Мартин, Джеймс Ньюкирк, Роберт Косс — Изд-во: Диалектика-Вильямс, 2004. — 752 с.