

ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ МЕТОДІВ СТВОРЕННЯ СЕРВІСІВ МОБІЛЬНИХ ДОДАТКІВ

На ринку мобільних платформ список лідируючих імен незмінний на протязі останніх трьох років. Зміни відбуваються виключно між цими лідерами. Таких лідерів четверо: IOS, Android, BlackBerry OS, Windows Mobile OS.

IOS – закрита операційна система від Apple. Випускається виключно для пристроїв, що розробляється компанією Apple. Розробка під IOS вимагає або наявності техніки Apple і програми розробки XCode, або використовувати сторонні сервіси для віддаленої компіляції додатків. Так само існує можливість розробки на Flash CS5.5 або інших середовищах флеш розробки за допомогою компілятора Flex, але варто враховувати, що продуктивність буде невисока. Публікація додатків так само не можлива без сертифікатів від віртуального магазину App Store, так само App Store є єдиним повністю легальним засобом дистрибуції додатків. Варто відзначити, що так само можливо поширювати додаток через піратські майданчики додатків, але це вимагає розблокування пристроїв, яка порушує гарантію, внаслідок чого, є непопулярною серед користувачів даної платформи. До плюсів цієї системи можна віднести стабільність і гарантовану роботу додатків на сучасних пристроях. Неналежні за якістю або контенту додатки не допускаються до поширення в App Store.

BlackBerry OS – операційна система від BlackBerry. Пристрої компанії BlackBerry націлені, в першу чергу, на категорію бізнесменів і корпоративних клієнтів. Має вкрай малу частку на ринку в світі, на травень 2013 року не більше 3 % (за даними IDC Worldwide Quarterly Mobile Phone Tracker). Працює тільки на пристроях компанії BlackBerry. Має можливість адаптації Android додатків під себе, але повна працездатність не гарантована. Windows 7 і Windows 8 (мобільні версії) – молода система від Microsoft. Для розробки потрібно Silverlight або XNA. Так само Microsoft випустила інструментарій розробника Windows Phone SDK для якого необхідні Visual Studio 2010 Express for Windows Phone і Expression Blend 4 для Windows Phone. На даний момент так само має малу частку ринку в Росії.

Android – відкрита операційна система від Google. Android дозволяє створювати Java додатки, що керують пристроєм через розроблені Google бібліотеки. Android Native Development Kit дозволяє портувати (але не налагоджувати) бібліотеки і компоненти додатків, написані на C та іншими мовами. Для дистрибуції додатків Google запустила онлайн-магазин Google Play. Для поширення програми через онлайн-магазин потребується платна обліковий запис, але так само можливо передавати програми будь-яким зручним для розробника способом як просто передавати інсталяційний. Арк файл так і через сторонні онлайн-магазини (наприклад Яндекс.Store).

На даний момент на ринку існує ряд онлайн-сервісів, що дозволяють створювати мобільні додатки за допомогою візуального редактора: AppInventor,

iBuildApp, Apps.ru, Kanchoo та інші. Такі засоби пропонують користувачеві інструмент для завдання зовнішнього вигляду програми та найпростішої логіки у вигляді вказівки переходів між екранами додатки. Для проектування додатків в розглянутих конструкторах використовуються різні графічні мови. Додатки будуються шляхом об'єднання стандартних компонент.

Недоліком таких сервісів є те, що вони не підтримують можливість завдання нетривіальною логікою додатка, наприклад: агрегацію даних, роботу із зовнішніми джерелами даних, авторизацію і т.д. Таким чином, функціонал більшості існуючих сервісів обмежується створенням інформаційних додатків, які не здатні на взаємодію з сервером або зовнішніми джерелами даних.

Крім того, багато з представлених на ринку сервісів дозволяють створювати додатки тільки для однієї певної мобільної платформи. Наприклад, проект Kanchoo дозволяє створювати додатки виключно для платформи iOS, а конструктор AppInventor – для платформи Android.

До того ж у процесі роботи в деяких конструкторах, наприклад, в AppInventor, потрібно безперервне підключення телефону до робочого комп'ютера. При від'єднанні телефону від комп'ютера до закінчення роботи над створенням додатка виконана робота не збережеться.

Конструктор[1] Buildanappic дозволяє розробляти програми для мобільної платформи iOS, Android, Blackberry і Mobile Web. Побудова додатки відбувається шляхом проходження ланцюжка з шести кроків. На кожному етапі користувачеві пропонується вибрати мету програми, необхідні елементи, кількість і вигляд сторінок в додатку і інші функції. BuildAnApp надає функціонал для використання електронної пошти і списку розсилки. Після створення додатку система пропонує завантажити кінцеве додаток на комп'ютер користувача. Цікавою особливістю є те, що власники мають можливість за допомогою BuildAnApp рекламувати свої створені програми, щоб заробити на них.

Web-додатки, що завантажуються в мобільний Web-браузер, відрізняються від вбудованих тим, що їх код пишеться з використанням Web-технологій (HTML, JavaScript і CSS), що не залежать від операційної системи пристрою. Немає необхідності вивчати різні мови програмування для кожного пристрою. HTML і JavaScript знайомі Web-розробникам щодо створення Web-сторінок для настільних браузерів.

У більшості випадків мобільні Web-браузери можуть візуалізувати ті ж самі Web-сторінки, але Web-сайти часто надають мобільні версії з меншим об'ємом інформації і більш швидкої завантаженням (через меншого розміру екрану і більш повільної мережі).

Для запуску Web-додатки користувач вводить URL-адресу в мобільний Web-браузер. Після цього завантажуються Web-сторінка, яка є точкою входу в Web-додаток. Web-додатки не поширюються через сховище додатків; вони є звичайними посиланнями, які можна включити в інші Web-сторінки, електронні повідомлення або навіть записати на папері.

Вбудовані і Web-додатки мають свої переваги і недоліки, будучи причиною багатьох дискусій про те, які програми краще. Вирішити цей спір можуть гібридні додатки, які намагаються об'єднати переваги обох типів мобільних додатків.

Гібридні програми, як і Web-додатки, програмуються з використанням Web-технологій, але пакуються як вбудовані додатки. Гібридне додаток можна написати відразу для декількох мобільних операційних систем з використанням мови програмування, знайомого багатьом розробникам. Оскільки гібридне додаток насправді є вбудованим, ви отримуєте доступ до функцій пристрою з JavaScript, що поки недоступно для Web-додатків. Гібридні програми можна поширювати і встановлювати через сховища додатків, подібно вбудованим.

PhoneGap [2] – це популярний набір інструментальних засобів для створення гібридних додатків. Він являє собою мобільну інфраструктуру з відкритими вихідними кодами, що містить JavaScript– інтерфейси для доступу до функцій різних пристроїв, наприклад, акселерометра і камери.

Є простий засіб релізу ваших додатків, за допомогою якого можна легко отримати бінарні файли вашої програми для всіх необхідних платформ. Документація просто чудова і дуже дружелюбна до початківцям розробникам. Можна розширювати SDK і додавати свої модулі, надавши їх реалізацію для кожної платформи рідною SDK.

З недоліків хочеться відзначити наступне: ваш додаток на всіх платформах буде виглядати однаково, рівно так, як ви це реалізуєте на HTML / CSS / JS; низька продуктивність, обмежена браузером ; погана доступність нативних функцій ОС (ніяких рідних діалогів, текстових повідомлень і специфічних контролів), відсутність спеціалізованої середовища розробки (підійдуть ваші улюблені інструменти для веб-програмування)

Розроблюваний конструктор повинен відповідати таким вимогам:

– Не вимагати від користувача знань і досвіду програмування для можливості створення додатку. Додаток повинен створюватися шляхом переміщення мишкою елементів палітри на екранну форму – все налаштовується і включається за допомогою різних елементів палітри і опцій.

– Підтримувати основні лідируючі на ринку мобільні операційні системи, такі як Android OS, Windows Mobile [3] і Apple iOS.

Загальна архітектура розробленої системи складається з декількох компонент і представлена. При розробці архітектури конструктора були використані результати, описані в роботах з проектування додатків А. Г. Федорова, С. Д. Кузнецова та книзі Мартіна Фаулера. Клієнтська частина складається з дизайнера і емулятора. Дизайнер пропонує засоби для моделювання інтерфейсу програми та завдання необхідної логіки. Емулятор забезпечує можливість перевірки коректності роботи створюваного мобільного додатку. Серверна частина, що включає веб-сервер і необхідні бази даних, виконує обробку необхідних даних і генерацію підсумкового файлу, який потім передається на клієнт і зберігається на комп'ютері користувача. За саму генерацію відповідає набір спеціально написаних генераторів додатків під різні платформи. У процесі роботи, сформований мобільний додаток взаємодіє з app сервером (сервером додатків).

Створення додатка відбувається шляхом перетягування на робочу область елементів палітри, таких як кнопка, зображення, текстове поле та інші. Аналогічним чином відбувається завдання логіки додатка і властивостей різних елементів. Наприклад, при додаванні нової кнопки, вказується назва форми на яку слід перейти при її натисканні і, при необхідності, умови для такого переходу.

Подібні властивості елементів можна задати або змінити в будь-який момент при проектуванні програми. Створене додаток можна протестувати за допомогою емулятора і потім згенерувати відповідний файл. Сформований файл (з розширенням `apk` для Android і з розширенням `har` для Windows Phone) автоматично зберігається на комп'ютері користувача.

У зв'язку з тим, що серверна частина програми використовується як репозиторій діаграм, клієнтська і серверна частини взаємодіють між собою за допомогою `post`-запитів (на відміну від `get` запитів, `post` запити передають всю необхідну інформацію в тілі запиту, а не в заголовку, і таким чином переважні для використання в системах, де необхідно передавати файли великої довжини).

Для можливості вільного доступу користувачів до розробленого конструктору, прототип дизайнера і емулятора викладений в «хмару» Amazon. В рамках даної дипломної роботи були реалізовані сайт, веб-сервер, `app` сервер, всі необхідні бази даних, а також забезпечена взаємозв'язок між різними компонентами системи (клієнтська частина, веб-сервер, генератори і сервер додатків). Розробка клієнтської [4] частини (дизайнера і емулятора) і генераторів не була частиною поставленого завдання і була проведена іншими людьми в рамках відповідних курсових і дипломних робіт. Тим не менше, їх опис наводиться тут для загального розуміння технології роботи реалізованого конструктора.

ЛІТЕРАТУРА

1. Burnstein, I. Practical software testing [Текст] / I. Burnstein. – Department of Computer Science. Illinois Institute of Technology Chicago, USA, 2002. – 732
2. Patton, R. Software Testing [Текст] / R. Patton. – Sams Publishing, USA, 2005 – 408
3. Fewster, M. Software Test. Automation Effective use of testexecution tools [Текст] / M. Fewster, D. Graham. – Biddies Ltd, Great Britain, 1994. – 591.
4. Myers, G. J. The Art of SoftwareTesting [Текст] / G. J. Myers. – John Wiley & Sons, Inc., Hoboken, New Jersey, Canada, 2004. – 255 с.

*Д. Ю. Лучкін,
студент V курсу,*

Харківський національний університет радіоелектроніки

*Т. Б. Шатовська,
доцент,*

Харківський національний університет радіоелектроніки

МЕТРИКИ ОЦІНКИ ЯКОСТІ ПРОГРАМНОГО КОДУ

Однією з тем у програмуванні є питання метрик коду програмного забезпечення. У великих програмних середовищах час від часу з'являються механізми підрахунку різних метрик. Хвилеподібний інтерес до теми так виглядає тому, що до цих пір в метриках не придумано головного – що з ними робити. Тобто навіть якщо якийсь інструмент дозволяє добре підрахувати деякі метрики, то що з цим робити далі найчастіше незрозуміло. Ця стаття – огляд найбільш відомих метрик коду програмного забезпечення.