

Лекція 8

Методології і технології розробки мобільних додатків

К.т.н. Талалаєв В.О.

План лекції

1. Призначення і типи мобільних додатків.
2. Методології розробки мобільних додатків.
3. Методики гнучкої розробки мобільних додатків.
4. Характеристика натівних середовищ розробки мобільних додатків.
5. Кросплатформенні середовища розробки мобільних додатків.
6. Популярні фреймворки розробки мобільних додатків.

Типізація мобільних додатків

1. Додатки-події, призначені для трансляції спортивних або інших подій.
2. Додатки-служби, які є аналогами сайтів, що відображають діяльність організацій.
3. Ігри, в тому числі розвиваючі та навчальні.
4. Інтернет-магазини, що розробляються для покупок в онлайн режимі.
5. Промо-додатки, що використовуються для реклами різних брендів.
6. Бізнес-додатки, що дозволяють оптимізувати процес роботи організації, забезпечуючи доступ до ділової інформації, інтеграцію з базами даних.
7. Системні функції, до яких належать додаткові налаштування і опції смартфона і його програмного забезпечення.
8. Навігаційні і пошукові сервіси, які застосовують GPSмодуль, що дозволяє використовувати телефон як повноцінний навігатор.
9. Мультимедійні додатки, що розширюють можливості телефону при роботі з відео і аудіо інформацією.
10. Соціальні мережі, що представляють собою онлайн-сервіси для спілкування, поширення інформації та організації соціальних взаємовідносин.
11. Контентні додатки і ін.

Мобільні додатки для смартфонів

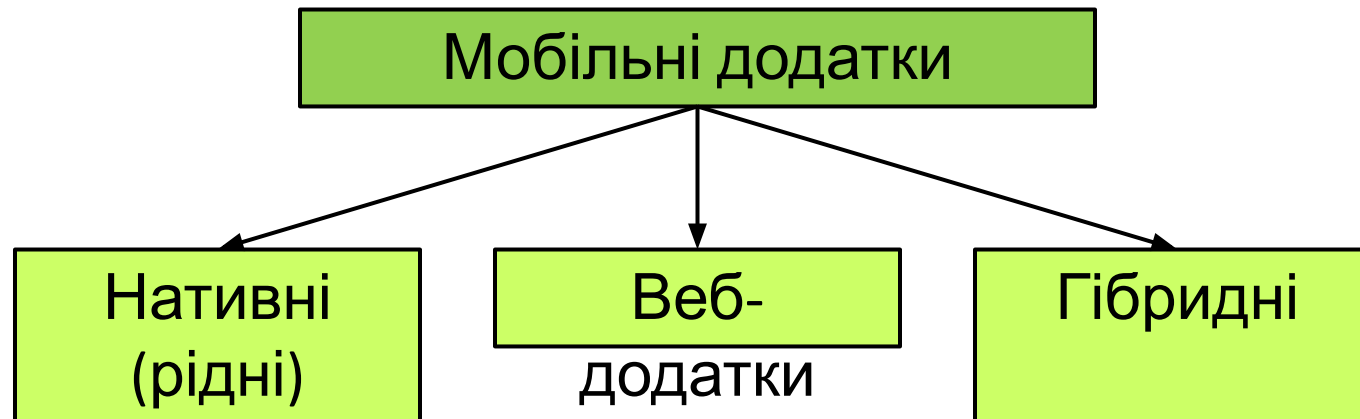
Розробка мобільних додатків нині є однією з найпопулярніших завдань в сфері інформаційних технологій. Мобільна розробка націлена на створення додатків, які можуть передбачати споживчі переваги; вирішувати завдання, алгоритм вирішення яких заздалегідь невідомі. Тепер додатки можуть здійснювати аналітичний аналіз інформації, отриманої з декількох джерел, допомагати користувачеві приймати рішення, контролювати процеси і вирішувати інші важливі завдання з найменшими тимчасовими і аналітичними витратами для нього самого. Це в свою чергу сприяє оптимізації бізнес-процесів, підвищення продуктивності та ефективності прийнятих рішень.

Мобільний додаток-спеціальна програма для смартфона, встановлена на тій чи іншій платформі, яка володіє певним функціоналом, що дозволяє виконувати певні дії в залежності від тих завдань, які необхідно вирішити. Це свого роду адаптер, що допомагає користувачеві взаємодіяти з різноманітною інформацією.

Типізація мобільних додатків

Перш ніж почати розробку мобільного додатку необхідно визначитися:

- хто буде користувачем (цільова аудиторія);
- для яких пристроїв призначене це додаток;
- в якому середовищі буде вестись розробка додатку;
- платформа, на якій додаток буде функціонувати;
- методологія і методи, які доцільно застосувати при розробці додатка.



Натівний додаток – це додаток, який розроблений на своїй (рідній) мові програмування для вибраної платформи (наприклад: *objective-c* для **ios**, *java* для **android**, *c#* для **windows phone**).

Причини використання:

працюють швидше і стабільніше, ніж додатки іншого типу;

дозволяють зняти функціональні обмеження браузерів з доступу до ресурсів пристрою.

Мобільна платформа надає інструментарій для розробників (SDK - software development kit), який дозволяє отримати доступ практично до всіх сервісів пристрої, а також призначений для спрощення процесу створення додатків.

Веб-додаток - це додаток, розроблене на HTML, JavaScript, CSS (Cascading Style Sheets - каскадні таблиці стилів) і вимагає для свого виконання встановленого і налаштованого браузера мобільного пристрою з виходом в Інтернет.

HTML застосовується для розмітки елементів інтерфейсу.

CSS описує візуальну складову і взаємне розташування віджетів і елементів управління.

Мова програмування JavaScript реалізує логіку програми.

Причини використання:

можливість повного або хоча б часткового повторного використання коду на різних платформах;

не пред'являють особливих вимог до графіки і використання апаратних засобів пристрою;

є величезний вибір інструментів, фреймворків, які прискорюють і спрощують процес розробки;

існування версії веб-сайту для настільного комп'ютера і є необхідність отримання доступу до через мобільний пристрій.

Гібридний додаток - це додаток, в якому частково використовується нативна функціональність, а частково - можливості веб-додатків.

(Від нативних додатків - можливість часткового доступу до ресурсів пристрою; від веб-додатків - підтримка HTML і робота в браузері).

Причини використання:

можна поширити його відразу на безліч платформ;

загальна продуктивністю і відгук інтерфейсу не є вирішальними;

можливість поширення (публікації) як готового продукту або тимчасового замітника до виходу нативного додатки (запустити процес маркетингу).

Порівняльний аналіз мобільних додатків

Порівняння типів додатків

Критерій \ Тип додатку	Нативні	Веб-додатки	Гібридні
Доступ до ресурсів	повний	отсутствует	ограниченный
Доступ до Інтернет	не обов'язковий	обов'язковий	обов'язковий
Установка	потребує	не потребує	потребує
Розповсюдження (публікація)	магазини додатків	не потребує	магазини додатків
Обновлення	повторне розміщення	не потребує	повторне розміщення
Платформи	розробка під кожену	крос-платформенні	крос-платформенні

Методології розробки мобільних додатків



Waterfall Model »(каскадна модель або« водоспад »)

Одна з найстаріших, передбачає послідовне проходження стадій, кожна з яких має завершитися повністю до початку наступної. У моделі Waterfall легко керувати проектом. Завдяки її жорсткості, розробка проходить швидко, вартість і термін заздалегідь визначені. Проте каскадна модель буде давати відмінний результат тільки в проектах з чітко і заздалегідь визначеними вимогами і способами їх реалізації. Немає можливості зробити крок назад, тестування починається тільки після того, як розробка завершена або майже завершена. Продукти, розроблені за цією моделлю без обґрунтованого її вибору, можуть мати недоліки (список вимог не можна скорегувати в будь-який момент), про які стає відомо лише в кінці через сувору послідовність дій. Вартість внесення змін висока, так як для її ініціалізації доводиться чекати завершення всього проекту. Проте, фіксована вартість часто переважає мінуси підходу.

Коли використовувати каскадну методологію?

- Тільки тоді, коли вимоги відомі, зрозумілі і зафіксовані. Суперечливих вимог немає.
- Немає проблем з доступністю програмістів потрібної кваліфікації.
- У відносно невеликих проектах.

Методології розробки мобільних додатків

«V-Model»

Успадкувала структуру «крок за кроком» від каскадної моделі. V-образна модель застосовна до систем, яким особливо важливо безперебійне функціонування. Особливістю моделі можна вважати те, що вона спрямована на ретельну перевірку і тестування продукту, що знаходиться вже на початкових стадіях проектування. Стадія тестування проводиться одночасно з відповідною стадією розробки, наприклад, під час кодування пишуться модульні тести.

Приклад - мобільний додаток для європейського стільникового оператора, який економить витрати на роумінг під час подорожей. Проект виконується за чітким ТЗ, але в нього включений значний етап тестування: зручності інтерфейсу, функціонального, навантажувального та в тому числі інтеграційного, яке повинно підтверджувати, що кілька компонентів від різних виробників разом працюють стабільно, неможлива крадіжка грошей і кредитів.

Коли використовувати V-модель?

- Якщо потрібно ретельне тестування продукту, то V-модель виправдає закладену в себе ідею: validation and verification.
- Для малих і середніх проектів, де вимоги чітко визначені і фіксовані.
- В умовах доступності інженерів необхідної кваліфікації, особливо тестувальників.



Методології розробки мобільних додатків

Incremental Model »(інкрементна модель)

У інкрементній моделі повні вимоги до системи діляться на різні збірки. Термінологія часто використовується для опису поетапної зборки ПО. Мають місце кілька циклів розробки, і разом вони складають життєвий цикл «мульти-водоспад». Цикл розділений на більш дрібні легко створювані модулі. Кожен модуль проходить через фази визначення вимог, проектування, кодування, впровадження та тестування. Процедура розробки за інкрементною моделлю передбачає випуск на першому великому етапі продукту в базовій функціональності, а потім вже послідовне додавання нових функцій, так званих «інкрементів». Процес триває до тих пір, поки не буде створена повна система.

Інкрементні моделі використовуються там, де окремі запити на зміну ясні, можуть бути легко формалізовані і реалізовані.

Коли використовувати інкрементну модель?

- Коли основні вимоги до системи чітко визначені і зрозумілі. У той же час деякі деталі можуть доопрацьовуватися з плином часу.
- Потрібно ранній вивід продукту на ринок.
- Є кілька ризикових функцій або цілей.



Методології розробки мобільних додатків

«RAD Model» (rapid application development model або швидка розробка додатків)

RAD-модель - різновид інкрементної моделі. У RAD-моделі компоненти або функції розробляються декількома висококваліфікованими командами паралельно, ніби кілька міні-проектів. Тимчасові рамки одного циклу жорстко обмежені. Створені модулі потім інтегруються в один робочий прототип. Синергія дозволяє дуже швидко надати клієнту для огляду щось робоче з метою отримання зворотного зв'язку і внесення змін.

Модель швидкої розробки додатків включає наступні фази:

- Бізнес-моделювання: визначення переліку інформаційних потоків між різними підрозділами.
- Моделювання даних: інформація, зібрана на попередньому етапі, використовується для визначення об'єктів і інших сутностей, необхідних для циркуляції інформації.
- Моделювання процесу: інформаційні потоки пов'язують об'єкти для досягнення цілей розробки.
- Збірка програми: використовуються засоби автоматичної збірки для перетворення моделей системи автоматичного проектування в код.
- Тестування: тестуються нові компоненти і інтерфейси.

Коли використовується RAD-модель?

Може використовуватися тільки при наявності висококваліфікованих і вузькоспеціалізованих архітекторів. Бюджет проекту великий, щоб оплатити цих фахівців разом з вартістю готових інструментів автоматизованої збірки. RAD-модель може бути обрана при впевненому знанні цільового бізнесу і необхідності термінового виробництва системи протягом 2-3 місяців.



Методології розробки мобільних додатків

Agile Model »(гнучка методологія розробки)

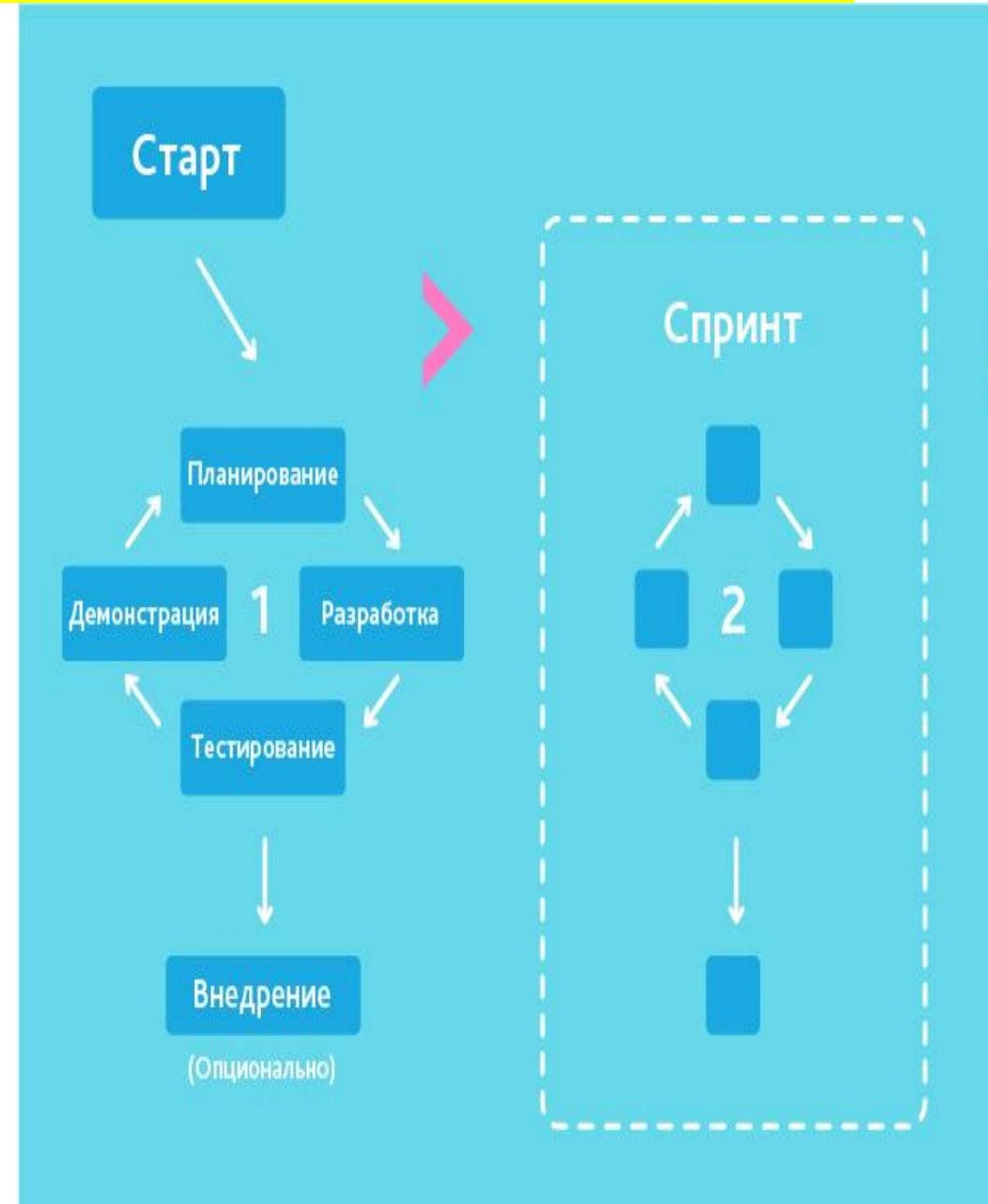
У «гнучкої» методології розробки після кожної ітерації замовник може спостерігати результат і розуміти, задовольняє він його чи ні. Це одна з переваг гнучкої моделі. До її недоліків відносять те, що через відсутність конкретних формулювань результатів складно оцінити трудовитрати і вартість, необхідні на розробку. Екстремальне програмування (XP) є одним з найбільш відомих застосувань гнучкої моделі на практиці.

В основі такого типу - нетривалі щоденні зустрічі - «Scrum» і регулярно повторювані зборів (раз в тиждень, раз на два тижні або раз на місяць), які називаються «Sprint». На щоденних нарадах учасники команди обговорюють:

- звіт про виконану роботу з моменту останнього Scrum'a;
- список завдань, які співробітник повинен виконати до наступних зборів;
- труднощі, що виникли в ході роботи.

Методологія підходить для великих або націлених на тривалий життєвий цикл проектів, постійно адаптуються до умов ринку. Відповідно, в процесі реалізації вимоги змінюються. Гнучка розробка найкраще підходить для цього психотипу керівників. Коли використовувати Agile?

- Коли потреби користувачів постійно змінюються в динамічному бізнесі.
- Зміни на Agile реалізуються за меншу ціну через часті інкременти.



Методології розробки мобільних додатків

«Iterative Model» (ітеративна або ітераційна модель)

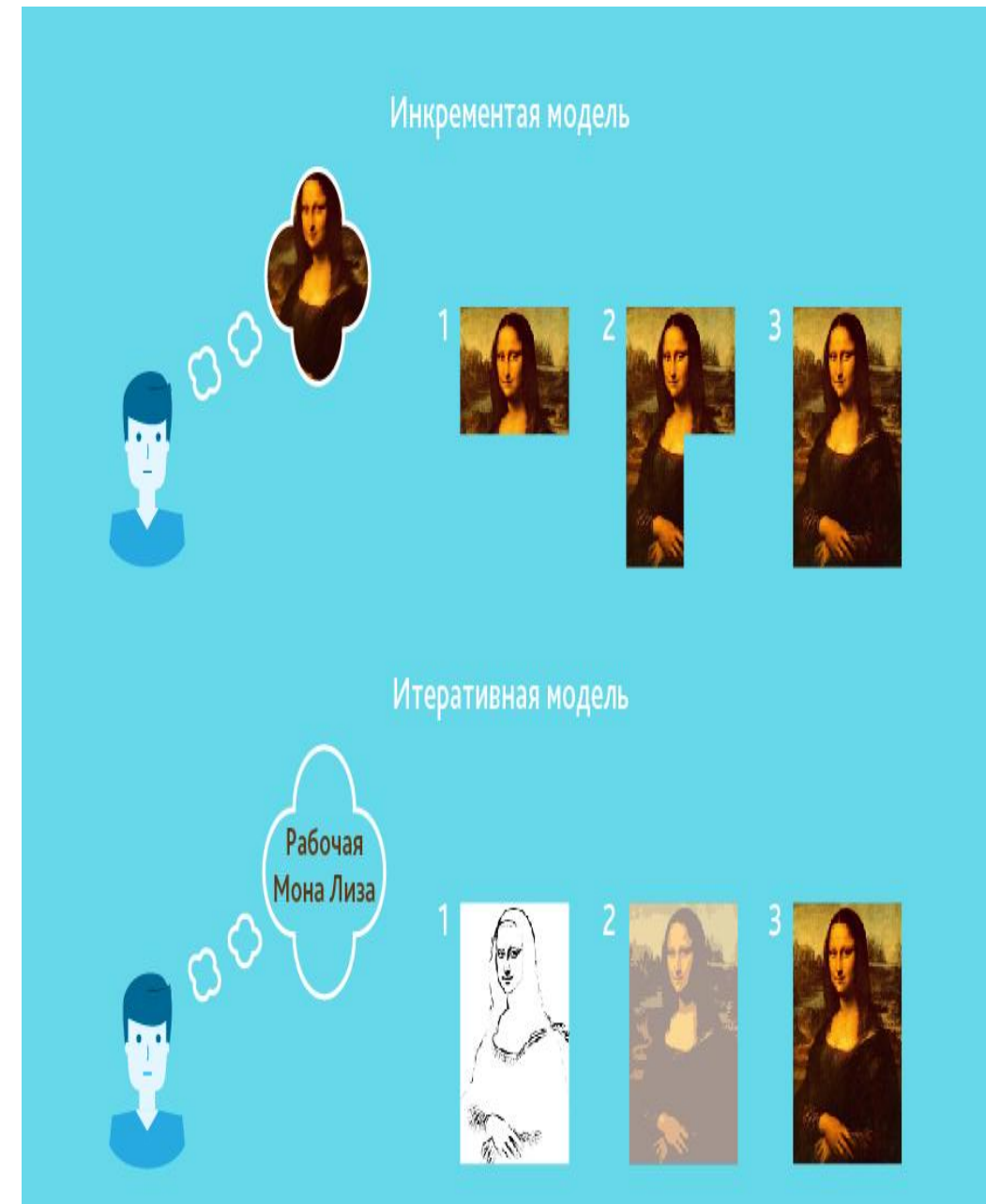
Ітераційна модель життєвого циклу не вимагає для початку повної специфікації вимог. Замість цього, створення починається з реалізації частини функціоналу, що стає базою для визначення подальших вимог. Цей процес повторюється. Версія може бути неідеальна, головне, щоб вона працювала. Розуміючи кінцеву мету, ми прагнемо до неї так, щоб кожен крок був результативним, а кожна версія - працездатна.

На діаграмі показана ітераційна «розробка» Мона Лізи. Як видно, в першій ітерації є лише начерк Джоконди, в другій - з'являються кольори, а третя ітерація додає деталей, насиченості і завершує процес. У інкрементній моделі функціонал продукту нарощується по шматочках, продукт складається з частин. На відміну від ітераційної моделі, кожен шматочок являє собою цілісний елемент.

Прикладом ітераційної розробки може служити розпізнавання голосу. Перші дослідження і підготовка наукового апарату почалися давно, на початку - в думках, потім - на папері. З кожною новою ітерацією якість розпізнавання поліпшувалося. Проте, ідеальне розпізнавання ще не досягнуто, отже, завдання ще не вирішена повністю.

Коли оптимально використовувати ітеративну модель?

- Вимоги до кінцевої системі заздалегідь чітко визначені і зрозумілі.
- Проект великий або дуже великий.
- Основне завдання повинна бути визначена, але деталі реалізації можуть еволюціонувати з плином часу.



Методології розробки мобільних додатків

Spiral Model »(спіральна модель)

«Спіральна модель» схожа на інкрементну, але з акцентом на аналіз ризиків. Вона добре працює для вирішення критично важливих бізнес-задач, коли невдача несумісна з діяльністю компанії, в умовах випуску нових продуктових лінійок, при необхідності наукових досліджень і практичної апробації.

Спіральна модель передбачає 4 етапи для кожного витка:

1. планування;
2. аналіз ризиків;
3. конструювання;
4. оцінка результату і при задовільній якості перехід до нового витка.

Ця модель не підійде для малих проектів, вона резонна для складних і дорогих, наприклад, таких, як розробка системи документообігу для банку, коли кожен наступний крок вимагає більшого аналізу для оцінки наслідків, ніж програмування. Державні проекти, в яких ми брали участь, починалися з підготовки експертним співтовариством дорогої концепції, яка аж ніяк не завжди марна, оскільки



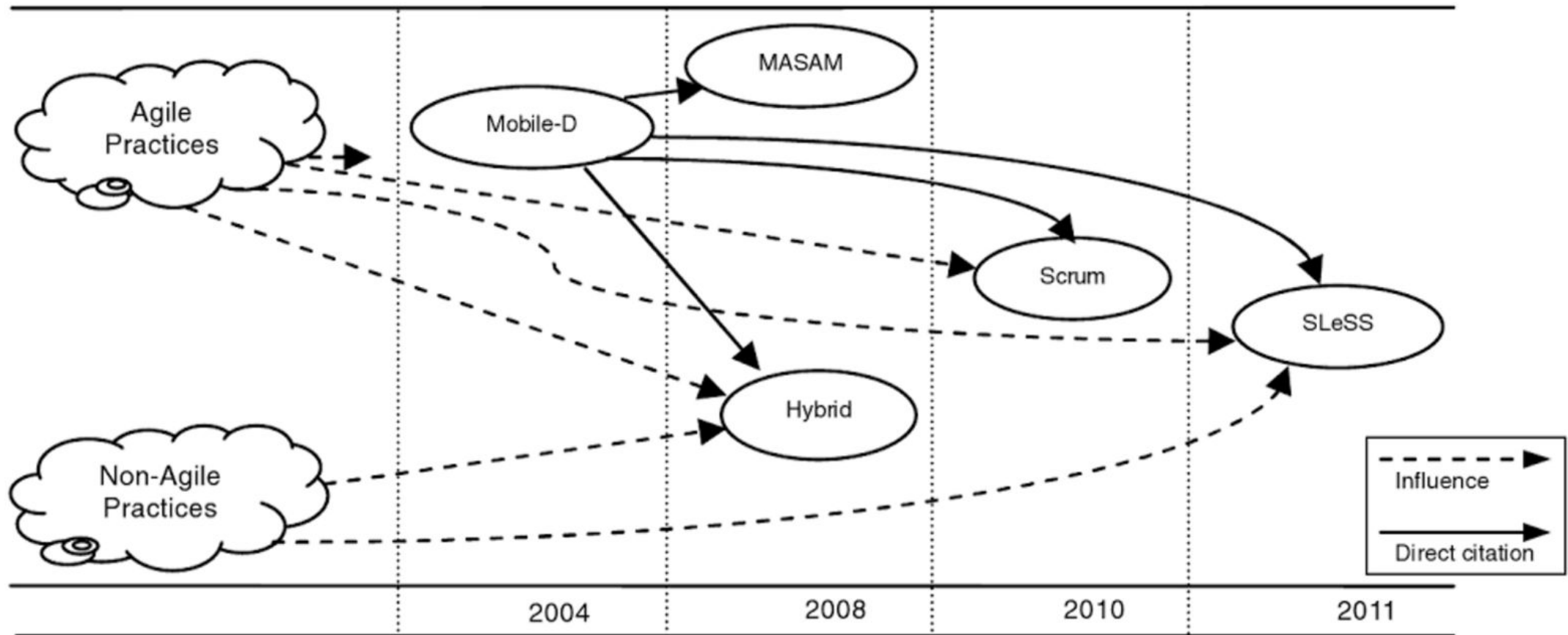
Особливості розробки мобільних додатків

- необхідність короткого часу доставки продукту на динамічний ринок, і подальших постійних оновлень;
- велика кількість користувачів з усього світу;
- складність у виявленні вимог до додатка, в тому числі з причини труднощів в ідентифікації стейкхолдерів;
- висока ймовірність змін потреб і очікувань користувачів, і відповідно необхідності вносити зміни по ходу розробки;
- високий темп технічної еволюції - нові пристрої, релізи ОС, ЯП, технології мобільного зв'язку & IoT, тощо.

Для того щоб відповідати цим характеристикам, знизити ступінь ризику, і впорядкувати процес мобільного розробки - широко застосовується методологія Agile з її адаптивним (допустимість частих змін), ітеративно-інкрементального (зворотний зв'язок із замовником на кожній ітерації, і множинні релізи), кооперативним (тісне співробітництво розробників, замовника і кінцевих користувачів) і простим (легко зрозуміти, змінювати і покращувати) підходом до розробки.

Технологія Agile дозволяє адаптувати процеси до нестійких потреб мобільного області. Agile забезпечує гнучкість для розуміння ринку, структурування продукту, і його випуску в короткі терміни.

За останні кілька років був розроблений ряд фреймворков, придатних для застосування в мобільній сфері: Mobile-D, MASAM, Hybrid, Scrum, SLeSS. Scrum і SLeSS є найбільш сучасними, при цьому в основі SLeSS лежить той же Scrum. Найперша спроба адаптувати Agile для мобільного розробки - Mobile-D - також викликає інтерес, перш за все з тієї причини, що включає в себе елементи інженерних практик XP.



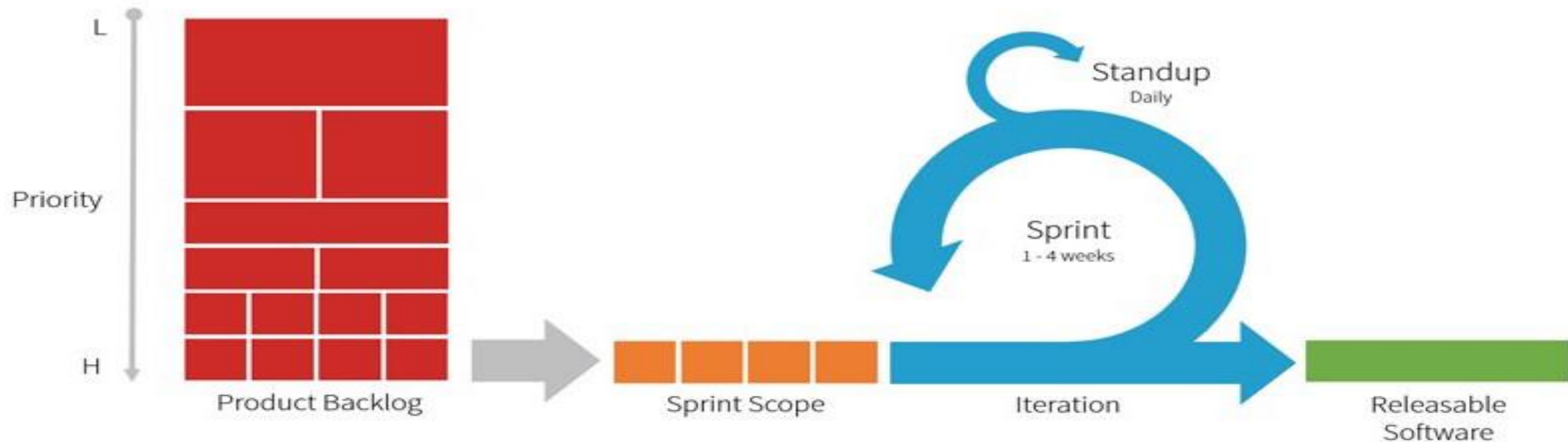
Scrum

Scrum - це одна з agile-методик, в якій створення продукту відбувається у вигляді серії ітерацій фіксованою тривалості. Методика робить акцент на якісному контролі процесу розробки, на управлінні завданнями в командному середовищі розробки.

У загальному випадку, Scrum передбачає роботу над проектом короткими ітераціями (зазвичай 2-4 тижні), кожна з яких є зменшеним варіантом проекту, і включає всі завдання, необхідні для видачі приросту по функціональності.

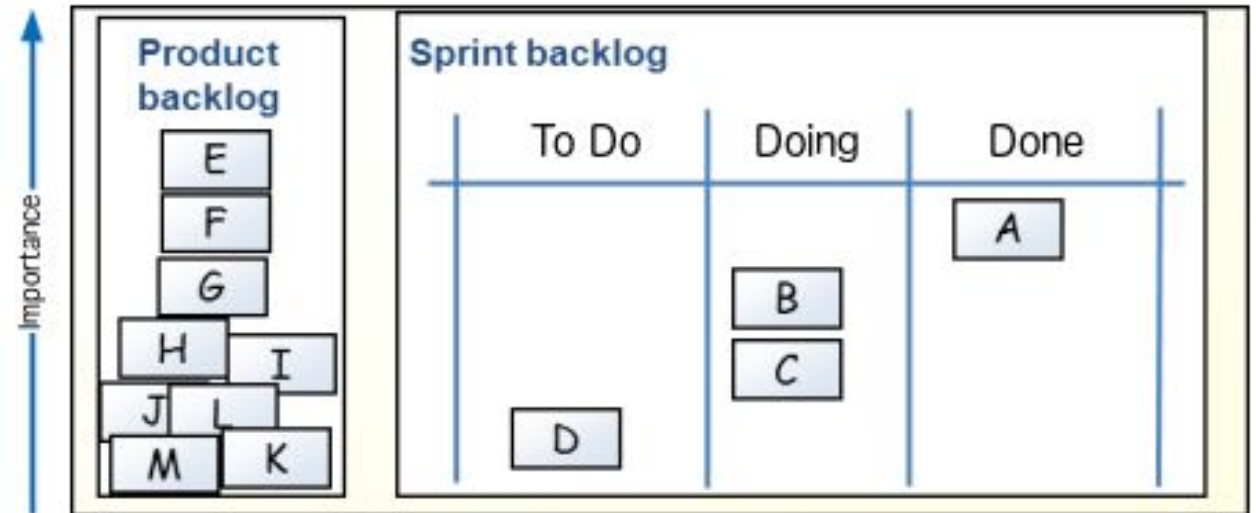
Кожен раз коли нова функція, модуль або оновлення завершено, здійснюється тестування. Завдяки тестуванню під час розробки, можливі баги виправляються вчасно, що в підсумку призводить до набагато більш стабільному продукту.

По закінченню ітерації замовник отримує нову версію продукту, і, якщо потрібно, проект або будь-яка його частина переглядається. Тут також враховується зворотний зв'язок від кінцевих користувачів. Потім цикл реалізації запускається знову. В результаті створюється рішення, яке максимально точно відповідає вимогам замовника та затребуване ринком.



Основні інструменти Scrum

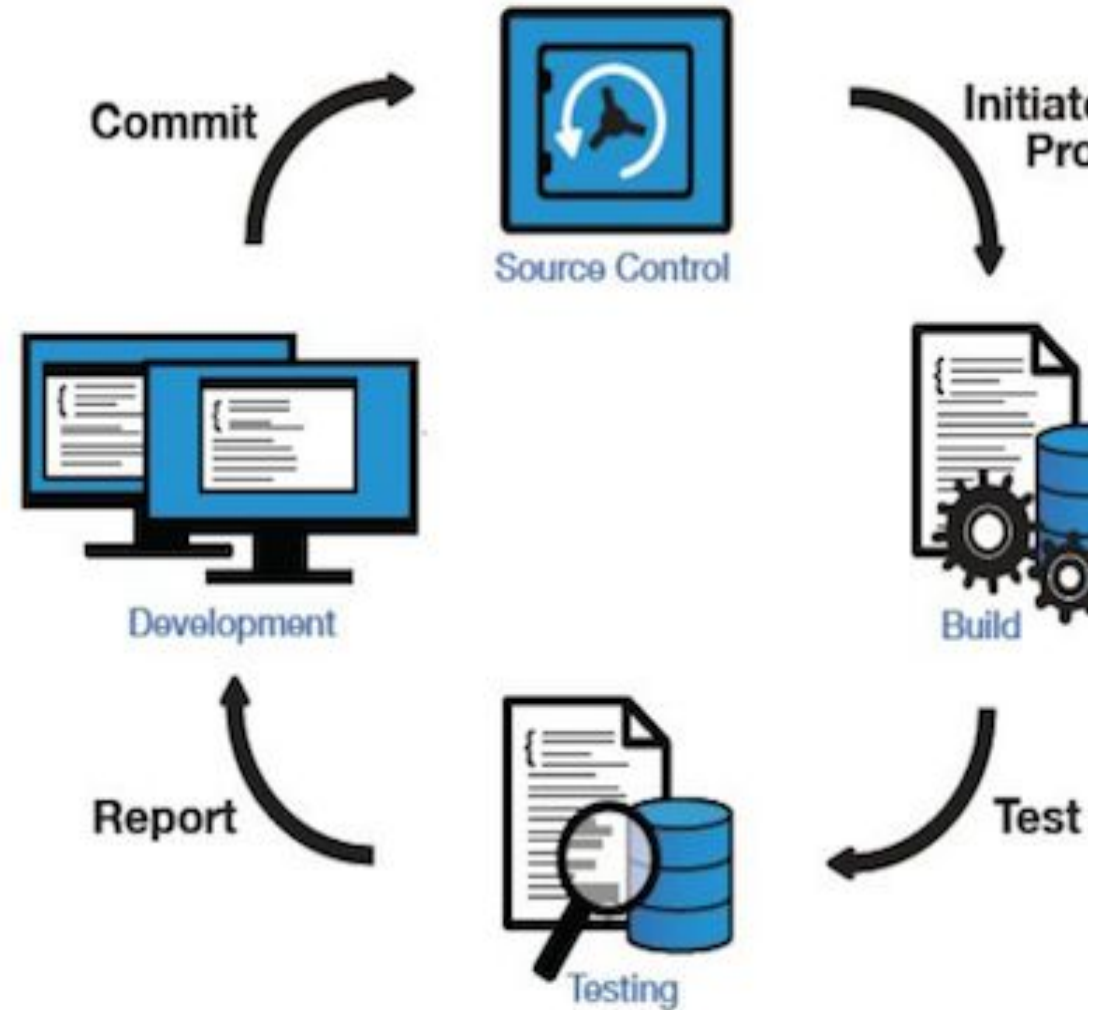
- Беклог продукте (Product Backlog);
- Історії користувачів (User Stories);
- Планування спринту (ітерація в Scrum);
- Беклог спринту (Sprint Backlog);
- Дошка задач (у вигляді фізичної або цифрової, наприклад, всередині JIRA)
- Діаграма згорання задач (Burndown Chart);
- Швидкі наради (стендапы);
- Огляд спринту (демо і ретроспектива).



XP

Agile-методика XP (Extreme Programming) сфокусована на інженерній стороні розвитку продукту. Це системний підхід до програмування. У той час як Scrum приділяє більше уваги управлінню і випуску, XP зосереджена на процесі виробництва. XP включає практики, що істотно поліпшують якість кінцевого продукту:

- Безперервна інтеграція (Continuous Integration (CI))
- Автоматизоване тестування;
- Рефакторинг;
- Рев'ю коду;
- Парне програмування;
- Стандарти кодування, і інші.



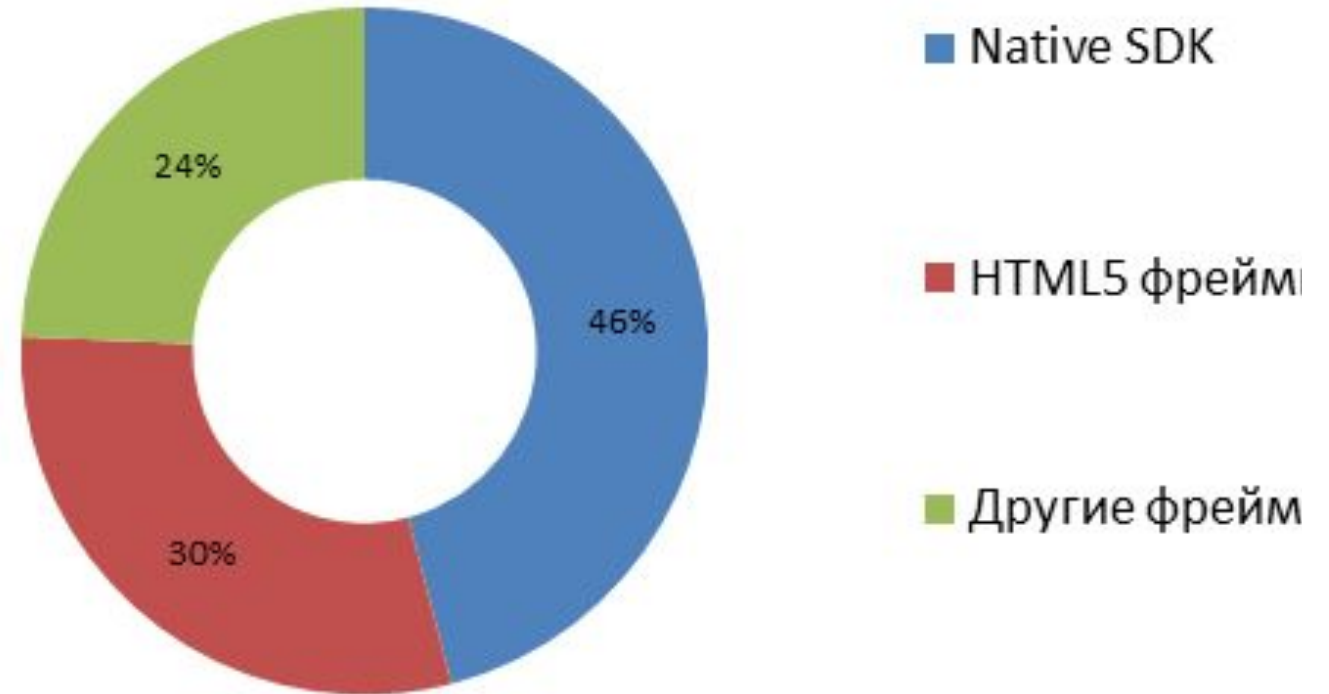
Середовища розробки натівних додатків

Природно, що вибір засобів залежить від призначення і складності додатку. Діаграма знизу показує, які середовища використовують розробники для роботи.

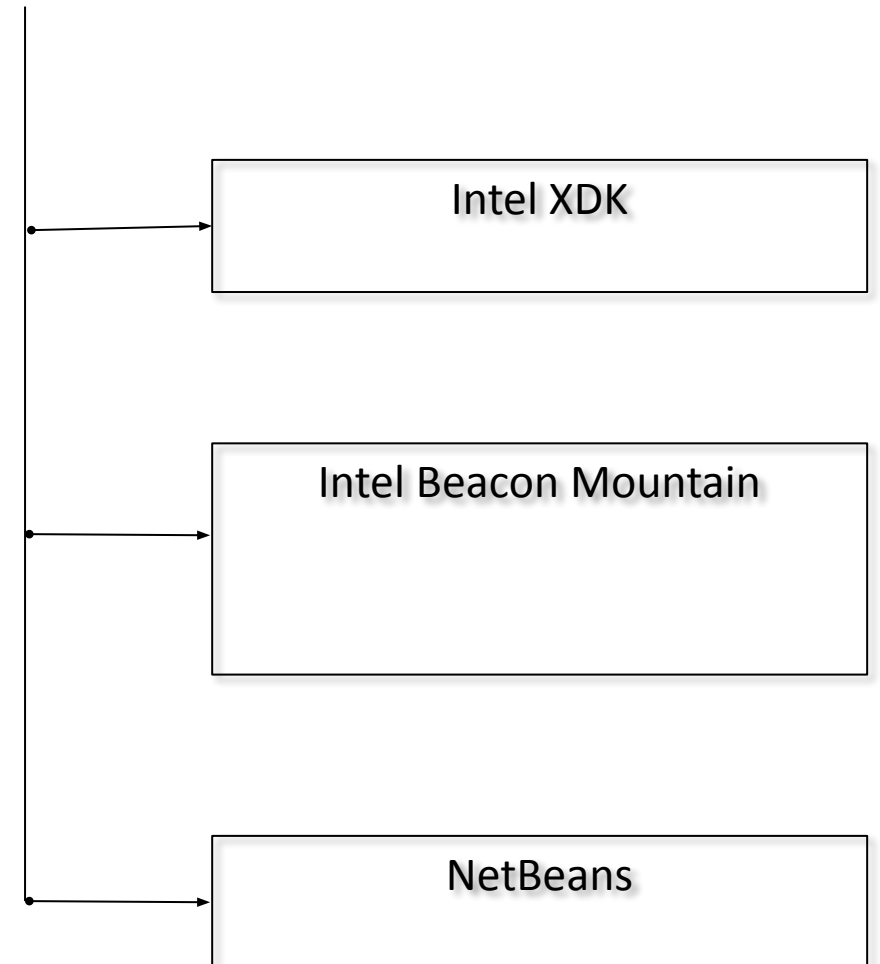
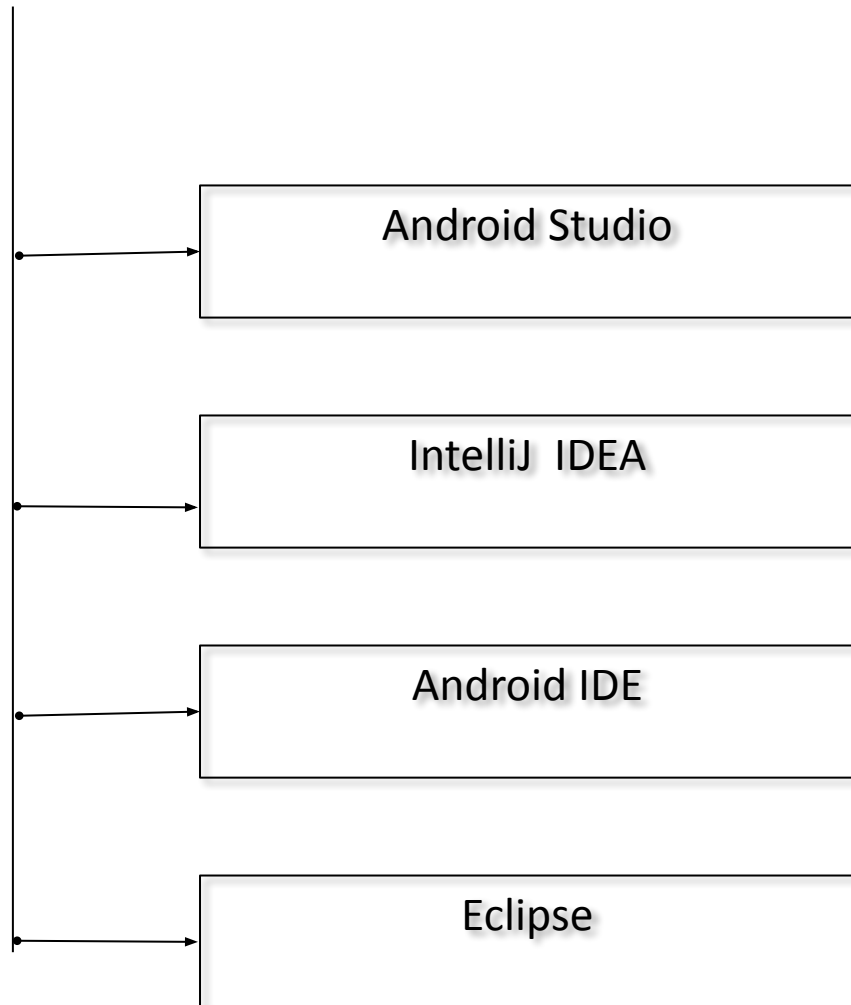
Середовища розробки дозволяють отримати повний доступ до можливостей операційної системи і компонентів телефону. Для створення програмного забезпечення вони використовують високопродуктивні мови програмування, саме тому вони дозволяють домогтися найвищої продуктивності, що критично для таких додатків, як ігри. Для різних платформ використовуються різні мови програмування:

- Для платформи Android використовується мова Java, проте можливо використовувати мови C / C ++, що дозволяє підвищити продуктивність на критичних ділянках коду;
- Для платформи iOS використовується мова ObjectiveC і Swift. Swift - нова мова програмування, представлений компанією Apple в 2014 році. Вона успадкувала більшість рис від ObjectiveC.

Разработчики

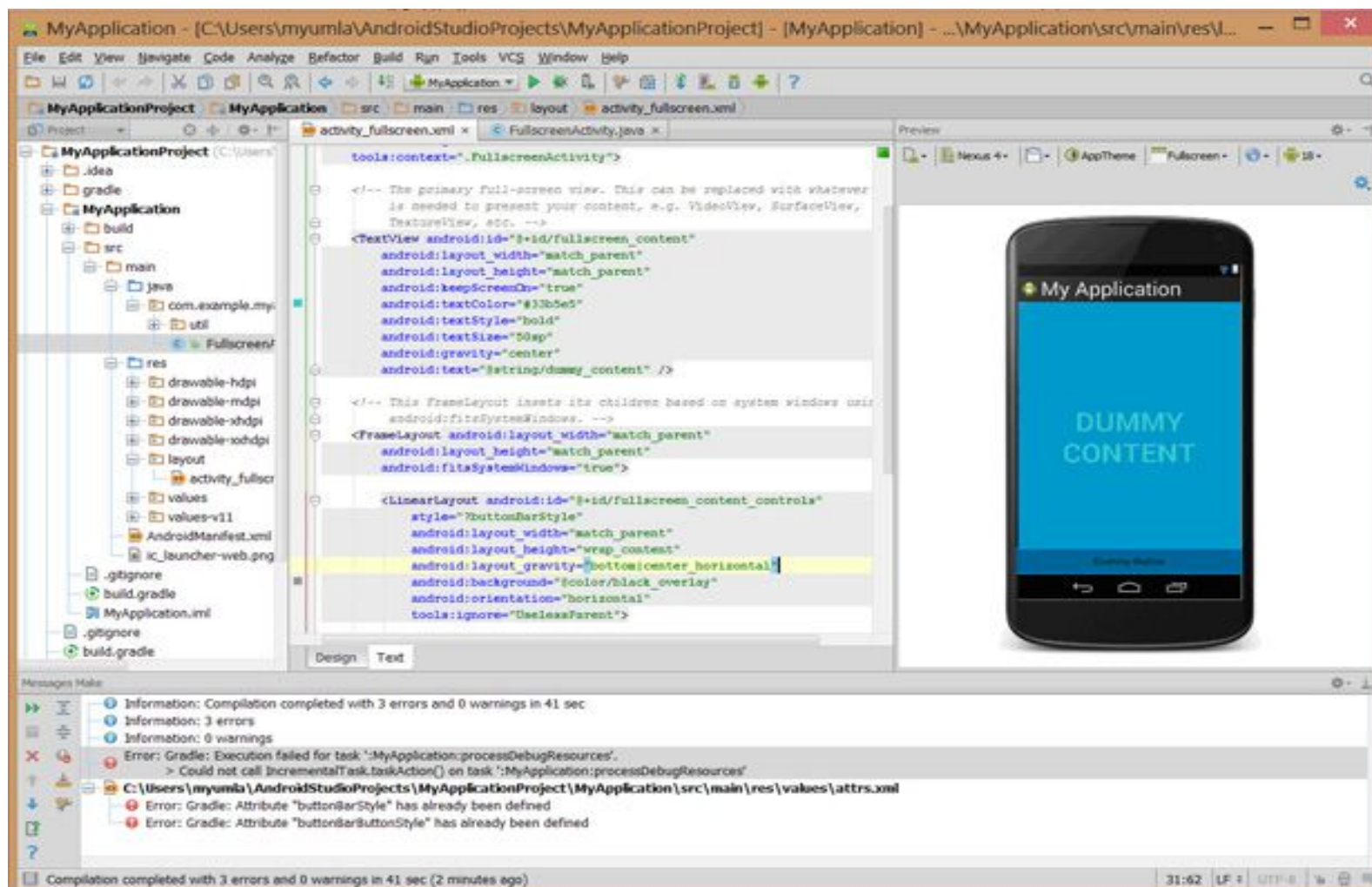


Розробка під Андроїд



Android Studio

Android Studio – інтегроване середовище розробки додатків для Андроїд від Google на основі IntelliJ IDEA



Android Studio

Середовище надає:

засоби для розробки додатків не тільки для смартфонів і планшетів, але і для носимих пристроїв на базі Android Wear, телевізорів (Android TV), очок Google Glass і автомобільних інформаційно-розважальних систем (Android Auto);

інструмент для автоматичного імпорту існуючого проекту Eclipse (ADT Plugin) в проект Android Studio;

засоби для спрощення тестування програм на сумісність з різними версіями платформи;

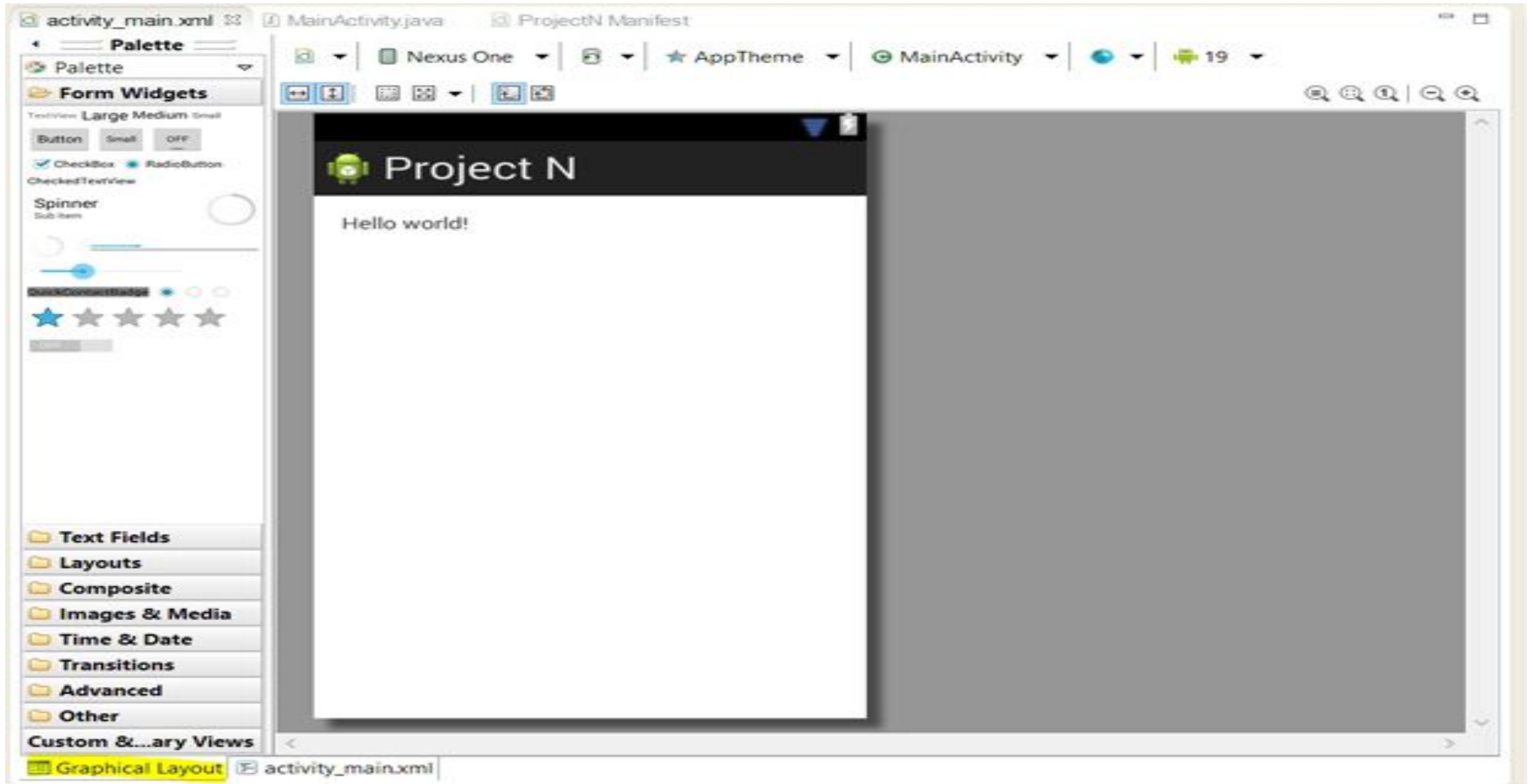
інструменти для проектування додатків, що працюють на пристроях з різними роздільними здатностями екранів (планшети, смартфони, ноутбуки, годинники, окуляри і т.п.)

Додаткові можливості

- збірка додатків, заснована на Gradle;
- специфічний рефакторинг і швидке виправлення дефектів;
- інструменти для пошуку проблем з продуктивністю, в експлуатації, з сумісністю версій та інших;
- утиліти для скорочення, оптимізації і «заплутування» коду, а також цифрового підпису додатків;
- майстри для створення загальних Android конструкцій і компонентів;
- редактор, який працює на багатьох розмірах екранів і роздільних здатностей, вікно попереднього перегляду, що показує запусканий додаток відразу на декількох пристроях і в реальному часі;
- інтерфейс перекладу на інші мови.

Android IDE

Android IDE – середовище розробки під Android, основане на Eclipse.



Android IDE

Містить:

засоби розробки (автодоповнення коду, перевірка помилок в реальному часі, рефакторинг і т.д.);

менеджер SDK для управління версіями SDK;

емулятор для розробки і тестування мобільних додатків без залучення реальних пристроїв;

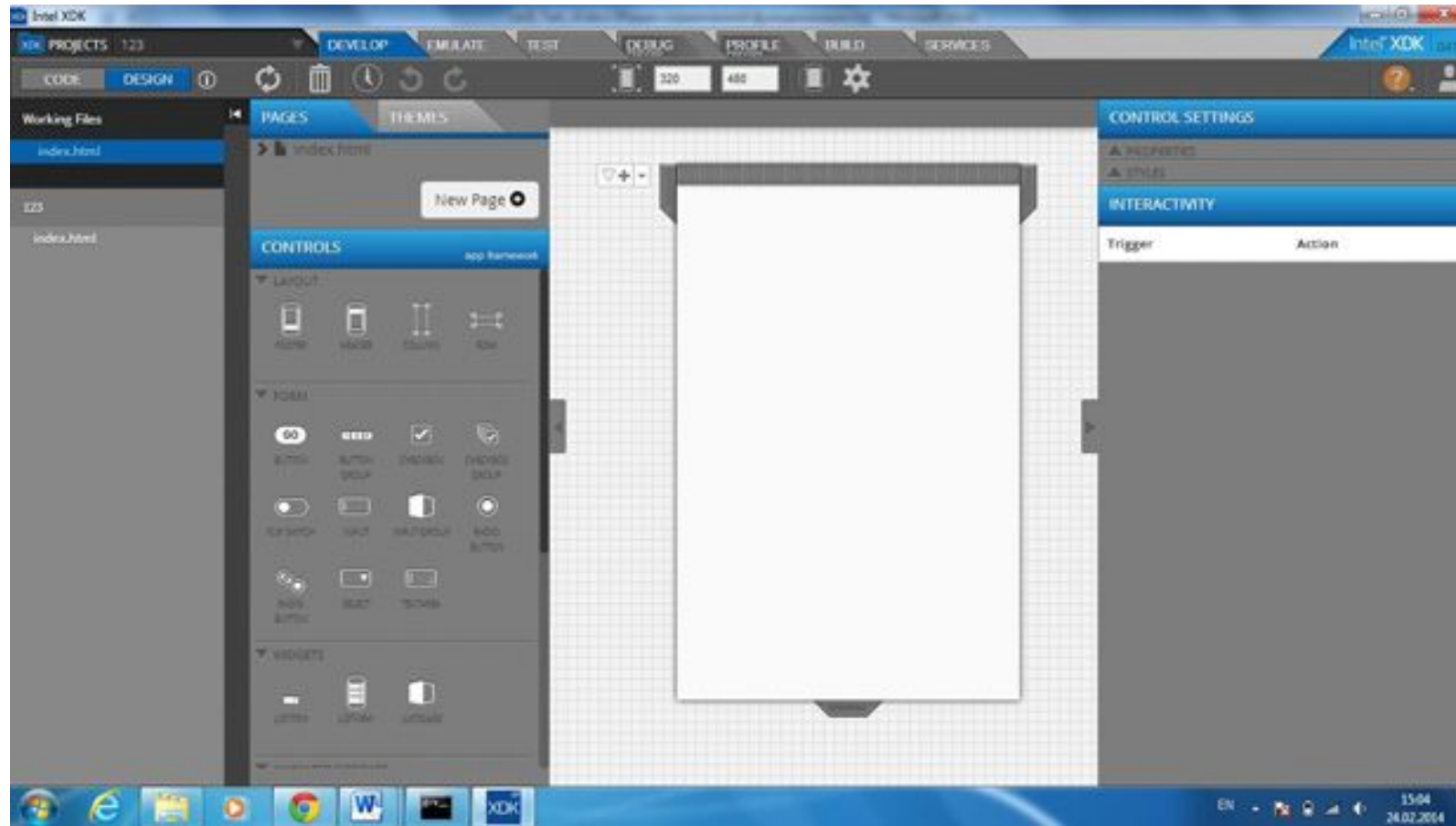
менеджери віртуальних пристроїв (інструменти для створення і управління віртуальними пристроями в Android (AVD) у вигляді окремих екземплярів емулятора);

інструмент, який надає графічний інтерфейс до кількох інструментів, утримуваних для аналізу та налагодження Android додатків;

засіб розробки на C / C ++ і Android NDK.

Intel XDK

Intel XDK дозволяє легко розробляти крос-платформенні мобільні додатки



Середовище надає:

- засоби розробки (редактор, емулятор пристроїв і відладчик);
- бібліотеку для користувача інтерфейсів Javascript, оптимізовану для мобільних додатків;
- додаток для тестування на пристроях;
- прикладний програмний інтерфейс для ігрових додатків з прискореною візуалізацією;
- систему Intel на базі хмари для підготовки версій додатків для більшості Інтернет-магазинів;
- засіб перенесення додатків iOS в середовище HTML5;
- набір функцій для взаємодії з операційною системою пристрою (підключення нативних плагінів, роботу з контактами телефону, камерою, геолокації, мульти-сенсорний, вбудованими відео- і аудіоплеєрами і т.д.).

Intel Beacon Mountain

Intel Beacon Mountain – набір інструментів, для проєкування, розробки, відладки і оптимізації додатків для Android (побудований на основі Android IDE).

(підтримує розробку для цільових платформ на основі процесорів Intel Atom и ARM).



Intel Beacon Mountain

Містить:

процесор віртуалізації для прискорення роботи емулятора в середовищі розробки (**Intel * Hardware Accelerated Execution Manager (Intel * HAXM)**);

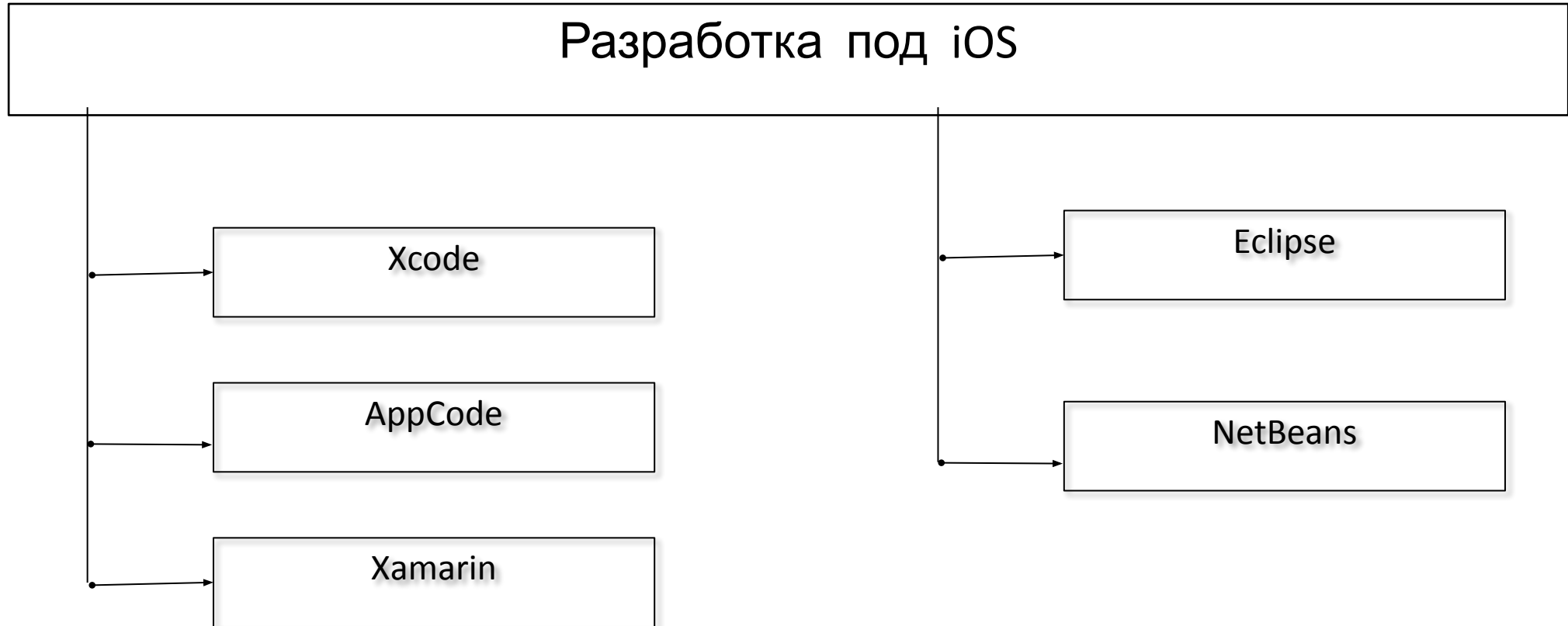
аналізатор продуктивності дозволяє оптимізувати завантаженість системи при використанні процедур OpenGL (**Intel * Graphics Performance Analyzers (Intel * GPA) System Analyzer**);

бібліотека оптимізованої обробки даних і зображень, що є частиною повної версії Intel IPP (**Intel * Integrated Performance Primitives (Intel * IPP) Preview**);

бібліотека шаблонів C ++ для створення масштабованих додатків і збільшення продуктивності за рахунок розпаралелювання (**Intel * Threading Building Blocks (Intel * TBB)**);

утиліта для установки оновлень, підтвердження статусу підписки на ПО, активації (**Intel * Software Manager**).

Разработка под iOS



Xcode

Xcode — це інтегроване середовище розробки відт *Apple (IDE)*.

включає:

редактор вихідного коду (підтримує мови C, C ++, Objective-C, Objective-C ++, Java, AppleScript, Python, Ruby, Perl):

графічний редактор для користувача інтерфейсу:

інтегрований відладчик:

емулятор різних пристроїв:

засіб розробки веб-додатків:

довідкова документація.



AppCode

AppCode - альтернативна среда розробки на Objective-C от компанії JetBrains.

Відмінності:

зручна навігація по коду;

поліпшена функція автодоповнення;

автоматизований рефакторинг;

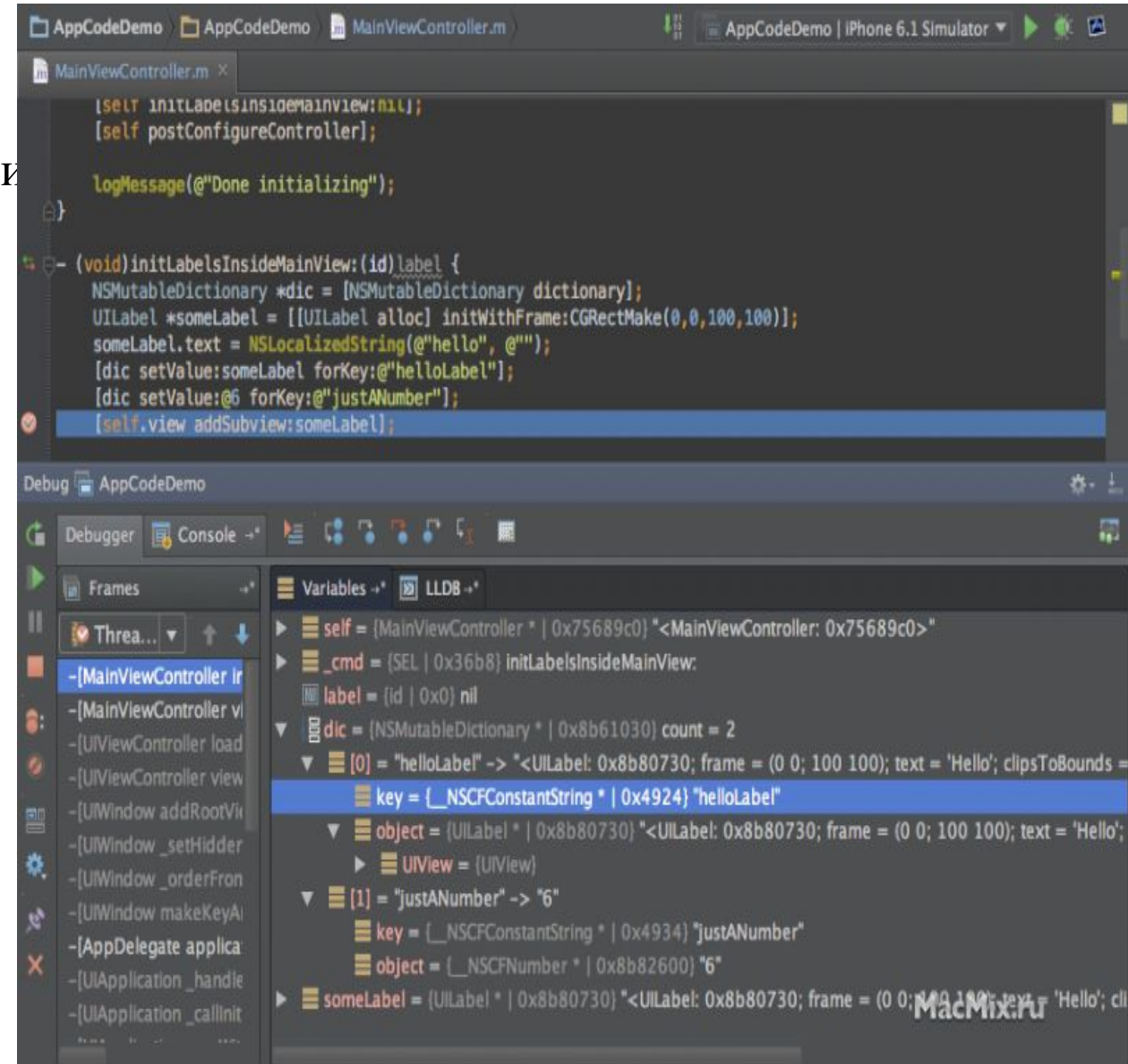
миттєвий аналізатор коду;

сумісність проектів з XCode і Interface Builder;

запуск додатків в емуляторі або прямо на пристрої;

покращений відладчик;

підтримка систем контролю версій.



Розробка під Windows Phone

Visual Studio Professional

Visual Studio Express

Visual Studio

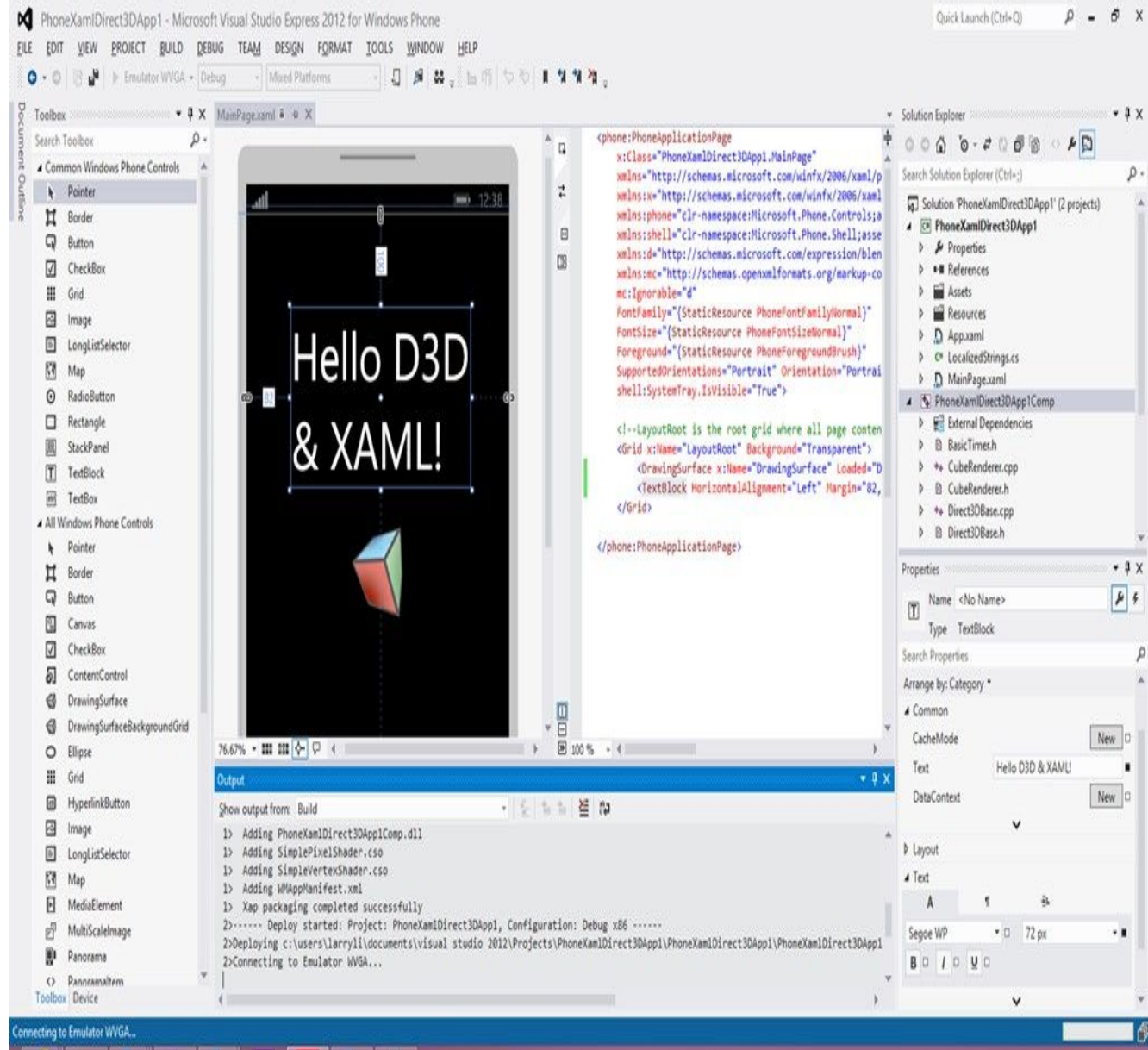
Visual Studio – это большая среда для разработки приложений под *Windows* и *Windows Phone*.

Має два варіанти комплектації, що містять Windows Phone SDK:

Visual Studio Professional;
Visual Studio Express.

Windows Phone SDK містить:

- Windows Phone SDK;
- Windows Phone Emulator;
- Windows Phone SDK Assemblies;
- Silverlight SDK and DRT (ПО для браузера для роботи з мультимедійними даними);
- XNA Game Studio (ПО для розробки ігор);
- Expression Blend (інтерактивний візуальний дизайнер для XAML);
- WCF Data Services Client (ПО для підтримки обміну даними через Інтернет);
- Microsoft Advertising SDK (для підтримки реклами).



Шаблони додатків

1) *Windows Phone Application* - простий діалоговий додаток, який має один головний екран, через який відбувається основна взаємодія з користувачем



2) *Windows Phone Pivot Application* – додаток з закладками, де заголовок кожної закладки зміст. Наприклад, календар, поштовий клієнт і налаштування телефону.



3) *Windows Phone Panorama Application* – додаток-панорама, в якому зони взаємодії з користувачем також розділені на панелі, але доступні вони через горизонтальну прокрутку (фонове зображення встановлено відразу на всю панораму, вона має загальний заголовок; контент сусідній панелі праворуч видно при відображенні поточної)

