

Проверка ВКР на чистоту на сайте «Главред»

10

баллов из 10
по шкале Главреда

476 предложений

5275 слов, 41855 знаков

0 стоп-слов.

Проверка грамотности фрагмента ВКР на сайте «Орфограммка»

Орфограммка

Грамотность

Красота

Качество

Титлы примечаний

Статистика

Для выбора технологии создания мобильного приложения для игротки вожатого был проведен опрос бойцов СПО «Мазстрон». В результате опроса выяснилось, что на телефонах почти в равной степени установлены как ОС Android, так и iOS (процентное соотношение: 40% к 60%). Поэтому необходимо создать гибридное приложение, которое будет обладать кроссплатформенностью, то есть способностью программного обеспечения работать на нескольких платформах. Такая разработка позволит охватить две операционные системы одним кодом. Это обеспечивает почти нативный подход к разработке благодаря интерфейсу визуализации с использованием собственных элементов управления.

Каждая операционная система мобильных устройств имеет свои технические возможности для запуска мекплатформенных программных продуктов. Рассмотрим решения создания кроссплатформенных приложений, которые существуют на данный момент.

Есть три варианта, которые можно использовать с технической точки зрения. Первое - использование WebView, которое имеется у всех операционных систем. Второй вариант - использование низкого уровня OpenGL/DirectX и языка C/C++ (Qt) или скомпилированного языка Dart (Flutter), с помощью которого можно получить высокую производительность, но не всегда очевидный Look and Feel. И третий подход - это применение Xamarin или ReactNative, которые обеспечат нативный пользовательский интерфейс и высокую производительность с минимальными расходами за счёт системного API верхнего уровня.

На рисунке 1.6 отражены архитектура, а также основные возможности и ограничения этих фреймворков.

Первый фреймворк, который необходимо рассмотреть: PhoneGap. Он основан на использовании WebView и не очень сложен в реализации, так как с помощью него разрабатывается небольшое нативное приложение, отображающее встроенный web-браузер и страницу HTML5. Все элементы интерфейса на странице имитируются под родные, так как отсутствуют нативные контроллеры и доступ к API. Чтобы получить доступ к системной функциональности необходимо иметь специальные плагины, через которые происходит нативная реализация на любой платформе с помощью JS-методов внутри web-браузера.

PhoneGap даёт возможность делить буквально весь код между платформами, но для этого требуется осуществление нативной части на ObjectiveC, Java и т.с. Если этот фреймворк применяется в известном Ionic, который даёт большую численность готовых плагинов для различных функций системы.

Но интерфейсы приложений, созданных на базе WebView, лишь делаются подобными на нативные с поддержкой HTML/CSS стилей.

Для создания программного продукта на PhoneGap необходимы знания работы с HTML, JavaScript, CSS, ObjectiveC, Java и неплохие навыки для интеграции нативной и кроссплатформенной частей. К сожалению, в приложениях с непростым интерфейсом такие одностраничные страницы будут тормозить и показывать не очень хороший результат.

Знаков: 6137 / 5358

Проверить грамотность (F7)

Проверка грамотности на сайте «Text.ru»

Проверка уникальности

Вы можете проверить данный текст на уникальность.

Запустить проверку

Проверка орфографии

В тексте найдено 6 ошибок:

- приемки-сдачи
- инкрементую
- инкрементирование

Подробнее

SEO-анализ текста

Всего символов: 5480

Заспамленность: 53%

Без пробелов: 4804

Вода: 12%

Количество слов: 669

Подробнее

Подсвечено: Ошибки в тексте

Главными преимуществами использования водопадной модели состоит в следующем:

- простота и понятность заказчику;
- постепенное прохождение стадий, завершающихся до начала следующей;
- на каждом этапе создается набор проектной документации, полной и согласованной заказчиком после приемки-сдачи;
- легкое планирование сроков выполнения работ и затрат на них.

Минусы водопадной модели заметны, когда нельзя определить конкретные требования или когда они нестабильны. При данной модели сложно вернуться хотя бы на один шаг назад для исправления недочетов, это увеличивает затраты на разработку и редактирование ошибок и срывает сроки разработки.

Рассмотрим другую модель управления – инкрементную.

В ее основе лежит инкрементирование, то есть процесс пошаговой реализации продукта за счёт постепенного расширения возможностей его функций. За основу берется заблаговременный полный набор требований. На первых фазах жизненного цикла создается архитектурное проектирование системы и вычисляется число необходимых инкрементов и их функций.

После на всех итерациях кодируются, тестируются и устанавливаются очередные инкременты. В первую очередь происходят этапы конструирования, тестирования и установки набора основных функций, которые создают базу продукта, обозначаются требования первой важности, которые нужны для создания успешного проекта.

Задача всех итераций – получение уже на ранних стадиях разработки работающей версии программного обеспечения с определенной функциональностью. После каждой итерации можно анализировать промежуточные результаты и реакцию на них заказчиков.

Такую модель стоит использовать при следующих обстоятельствах:

- если большинство требований и функций можно сформулировать заранее, но они должны появиться не сразу;
- если необходимо выполнить быструю поставку продукта на рынок, хотя бы с базовыми свойствами;
- если проект долгосрочный, то есть на него предусмотрено большое время на разработку, минимум год;
- при разработке продуктов с низкой или средней степенью риска;
- при проекте, использующем новые технологии.

Инкрементная модель подходит для проектов, в которых точное техническое задание прописано уже на старте, а продукт должен быстро выйти на рынок.

Третья модель, которая является комбинацией предыдущих двух, называется спиральная. Её главная особенность заключается в том, что она учитывает риски (например, отсутствие необходимых ресурсов, потеря актуальности приложения, риск сорвать сроки и т.п.). Такая модель помогает внедрить элементы разработки программного обеспечения из нескольких моделей процессов для программного проекта на основе

Вы можете повысить уникальность текста на нашей Бирже рерайтинга.

Повысить уникальность

Версии текста:

Несколько секунд назад (UTC +03:00)

Уникальность	N/A	Орфография	6
Всего символов	5480	Заспамленность	53%
Без пробелов	4804	Вода	12%
Количество слов	669		

Доступность проверки

Результат проверки текста доступен только вам