УДК – 004.04

# Кодогенерация как способ оптимизации времени в отделе мобильной разработки

Магистрант 1 курса И. В. Кислюк

Руководитель - старший преподаватель С. В. Одиночкина

*Университет ИТМО, Санкт-Петербург*

*igorkislyuk@icloud.com*

В разработке ПО важным моментом становится поиск баланса между ключевыми параметрами, к которым относят требования, временные и финансовые затраты. Выигрыш в одном параметре позволяет получить преимущества в остальных. Например, оптимизация затрат времени команды на разработку сократит финансовые расходы. Оптимизация может проводиться на различных уровнях проекта – на уровне управления, через смену модели разработки, или на уровне работы с программным кодом, путем автоматизации процесса код-ревью, генерации кода [2]. Кодогенерация – процесс генерации кода на основе определенных данных. Различают несколько видов кодогенерации: на основе более высокоуровневого кода (использование расширений LINQ в языке высокого уровня C#), на основе метаданных (подход «Database First» в Entity Framework) и на основе шаблонов (формирование страниц ответа в серверном JavaScript «ejs») [1].

Современные мобильные приложения представляют из себя «тонких клиентов» в рамках клиент-серверной архитектуре, делая больший упор на отзывчивость и красоту пользовательского интерфейса. Кодогенерация может и должна применяться с обеих аспектов мобильных приложений. Существует множество открытых решений по генерации файлов мультимедиа и локализации компонентов приложений [5]. В процессе поиска не было найдено решение для оптимизации взаимодействия мобильного клиента с сервером.

На основании этого были предъявлены следующие требования к создаваемому приложению: получение моделей запроса и ответа с возможностью адаптации под любую платформу и язык программирования, а также генерация документации для разработчиков. Согласно требованиям, была разработана слоистая архитектура генератора, которая состоит из моделей описания запросов и ответов сервера, промежуточных моделей для реализации полиморфных интерфейсов и результирующих моделей для отображения их в файлы программного кода. После определения требований была реализована генерация кода и успешно встроена в новые проекты компании. По результатам использования генератора на проектах, общее время реализации взаимодействия с сервером суммарно на платформах сократилось на 9-13%, что дает основание считать данную оптимизацию выигрышной.

Представленный вид кодогенерации позволяет оптимизировать временные затраты в проектах, почти устраняя дублирование реализации клиент-серверного взаимодействия на различных мобильных платформах. Среди недостатков стоит отметить первоначальную сложность интегрирования в новые проекты, невозможность полноценного использования в существующих проектах. Кодогенерация призвана упростить целиком разработку мобильных бизнес-приложений. Примерами таких проектов являются React Native и Apache Cordova.

**ЛИТЕРАТУРА**

1. Подходы к кодогенерации [Электронный ресурс]. – Режим доступа: https://habrahabr.ru/post/23858/ свободный. Язык рус. (дата обращения 04.02.2018)
2. Research and Application of Code Automatic Generation Algorithm Based on Structured Flowchart.– Harbin Institute of Technology, 2011.– 12 с.
3. React Native [Электронный ресурс]. – Режим доступа: https://facebook.github.io/react-native/ свободный. Язык англ. (дата обращения 04.02.2018)
4. Apache Cordova [Электронный ресурс]. – Режим доступа: https://cordova.apache.org свободный. Язык англ. (дата обращения 04.02.2018)
5. Генератор ресурсов интерфейса [Электронный ресурс]. – Режим доступа: https://github.com/SwiftGen/SwiftGen свободный. Язык англ. (дата обращения 04.02.2018)